STF to CTDB and CTDB to STF Conversions

SEDRIS Technology Conference
August 23, 2002

Vancouver, British Columbia

Presenters

Andrew Tosh (Acusoft) *tosh@acusoft.com*

Kevin Wertman (SAIC) *kevin.m.wertman@saic.com*

Introduction

- DESCRIPTION
  - Compact Terrain Database (CTDB) is an optimized run-time format used by the ModSAF and OneSAF applications. Two complementary converter applications that respectively take STFs to the CTDB format, and convert existing CTDB databases to STF will be described. The tutorial will cover the capabilities and use of both applications. The tutorial on the STF to CTDB conversion will cover the basics of the CTDB format, what the compiler expects to find in a SEDRIS transmittal in order to produce a useful CTDB database, and the variety of terrain data types that can be converted to CTDB. The CTDB to STF portion of the tutorial will cover the conversion process and the mapping of CTDB data to STF, and how SEDRIS-based analysis and visualization tools are utilized to examine the content of CTDB databases.
- WHO SHOULD ATTEND
  - This tutorial is intended for developers or users of ModSAF, users of other Computer Generated Forces (CGF) applications, and those interested in converting terrain data to CTDB through SEDRIS. One goal is to educate ModSAF users on the CTDB format.
- PREREQUISITE
  - Some familiarity with CTDB and ModSAF will be helpful. Prior attendance at either the "Introduction to SEDRIS for Managers" or the "Fundamentally SEDRIS: The Technology Components" tutorial is recommended.
- WHAT TO EXPECT
  - The attendees will learn what type of data the STF to CTDB converter will expect, how to use both applications, and how the conversions have been implemented.

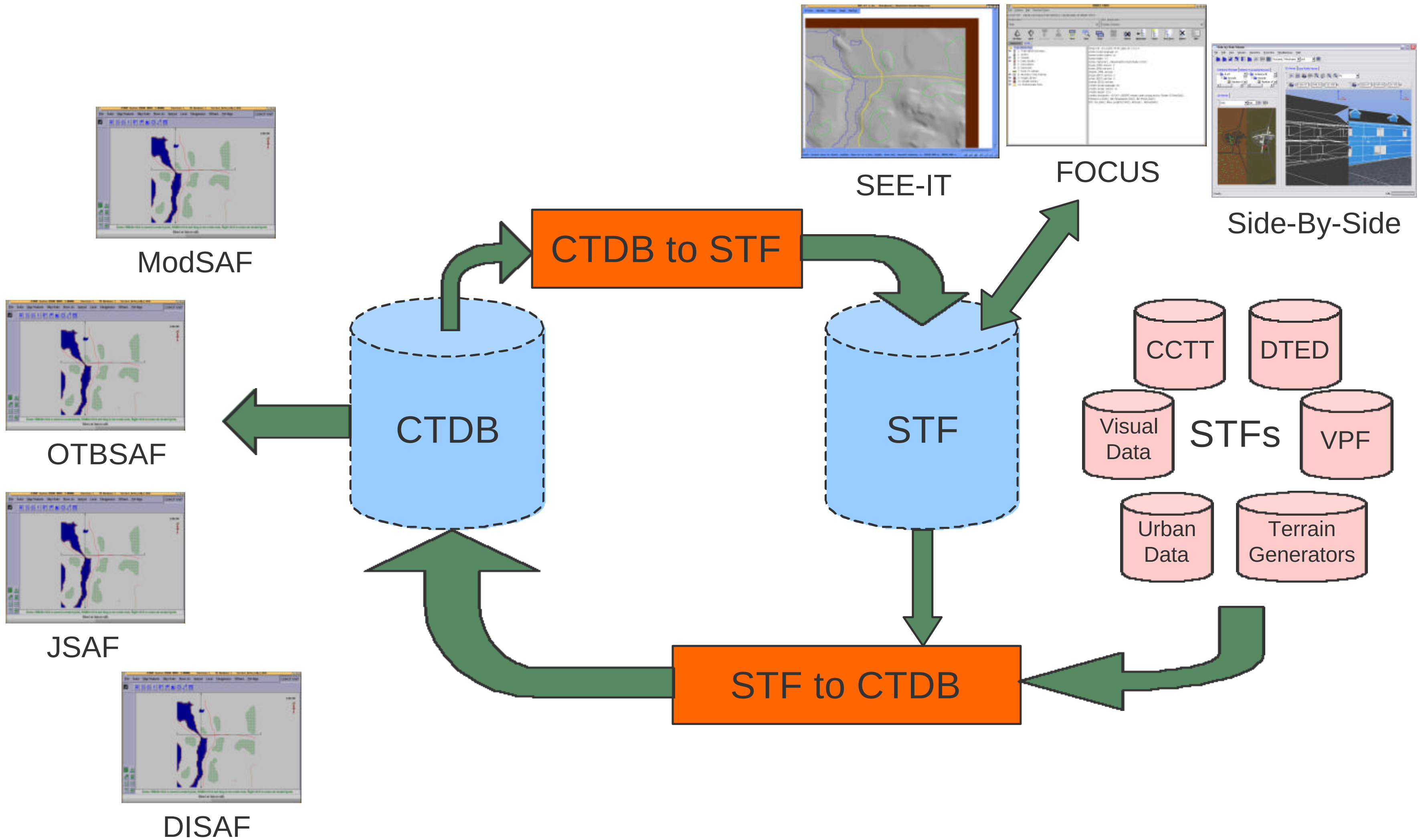
Prerequisite

- To get the most out of this tutorial, we assume you know the following information as a prerequisite to this session:
  - A solid understanding of the SEDRIS technology components (including the DRM and SRM) and how they fit together. We assume you have attended:
    - *"Introduction to SEDRIS for Managers"* or
    - *"Fundamentally SEDRIS: The technology components"*
  - Familiarity with CTDB and ModSAF will be helpful.
  - An interest in using tools to convert CTDB to and from STF.

Agenda

- Introduction to the CTDB format
- CTDB to STF
- STF to CTDB
- Questions

CTDB Process Diagram



CTDB Format: Agenda

- Introduction to the CTDB format
  - Basic Information
  - Polygon Attribute Table
  - Terrain and Soil Representation
  - Feature Representation
    - Physical Features
    - Abstract Features
  - MES
- CTDB to STF
- STF to CTDB
- Questions

CTDB Format: Basic Information

- Compact Terrain Database (CTDB)
  - Very space efficient format
- Was created for ModSAF and continues to be used with OTBSAF, JointSAF, DISAF
- Coordinate System
  - Originally UTM based
  - Expanded to support Global Coordinate System (GCS)
    - Single Cell or Multi Cell
    - GTRS

CTDB Format: Polygon Attribute Table

- PAT is a 2 dimensional lookup table.
  - 4 default columns: Mobility Word, Soil Type, Surface Material Composition, and Surface Wetness Condition.
  - Each row has a unique combination of values for the different columns.
  - Additional columns, which store FACC attributes, can be added.
- 1st column, the mobility word is a packed word containing the following information:
  - SIMNET Trafficabilty Code
  - CCTT Trafficabilty Code
  - Water Flag
  - No-Go Flag
  - Road Flag

CTDB Format: Terrain and Soil

- Terrain can be stored as
  - Elevation Grid Posts
    - With accompanying soil grid that references the PAT for each elevation post.
    - Can have polygonal microterrain stored with the features for higher resolution elevation information.
  - Triangulated Irregular Networks (TIN)
    - Stored as polygonal data with topological information to make elevation lookups faster.
    - Each polygon references the Polygon Attribute Table (PAT) for soil and mobility information.

CTDB Format: Features

- Physical Features
  - Stored in patch structures
    - Patches are generally 500x500m squares
      - Patch size can vary based on post spacing
      - 4 or 5 posts per patch
  - Generally used for intervisibility and impedance of movement
  - Certain features used for route-planning
- Abstract Features
  - Stored in quad tree structure
  - Used for some route-planning and general information

CTDB Format: Physical Features

- Volume Models (Buildings, obstacles)
  - Stored in CTDB as a set of 3D vertices representing the roofline of the volume.
  - SAF "drops down" walls to the terrain for each pair of vertices.
  - CTDB limits number of vertices to 15.

CTDB Format: Physical Features

- Tree lines
  - Stored as a linear set of points.
  - Additional information:
    - Trunk Radius
    - Foliage Height
    - Fullness
    - Total Height
  - Affects intervisibility and acts as a obstacle to vehicle movement.
- Individual Trees
  - Stored as tree line with only one point.
- Tree Canopies
  - Stored as a set of polygons representing the roof of the canopy.
  - The fullness of the canopy is also stored to represent the density of the forest area.
  - Canopy polygons are used in intervisibility calculations.

CTDB Format: Physical Features

- Laid Linears
  - Roads and Rivers
  - Stored as a set of 2D locations
  - The width of linear is also stored with the feature
  - Laid Linears are used in route planning

CTDB Format: Abstract Features

- Canopies
  - Tree Canopy Footprint
  - Stores impenetrable bit that indicates whether vehicles can penetrate the canopy
  - The Abstract Canopy Feature is used for route planning
- Soil Defragmentation Areas
  - Bounding footprint of an area with a uniform soil type
  - Stores reference into Polygon Attribute Table to indicate what the soil type the terrain within the defrag should have

CTDB Format: Abstract Features

- Steep Slopes
  - Bounding footprint of an area with a steep slope
  - Stores the degree of the slope angle
- Political Boundary
  - Set of points representing a political boundary such as country borders or city limits
- Labels
  - Single point and a string value representing a map label

CTDB Format: Abstract Features

- Railroads and Pipelines
  - Both stored as linear sets of points
- Tactical Sign
  - Area identified as tactically dangerous
- Off road segment
  - Allows for traffic networks that don’t follow any roads
  - Used for route planning

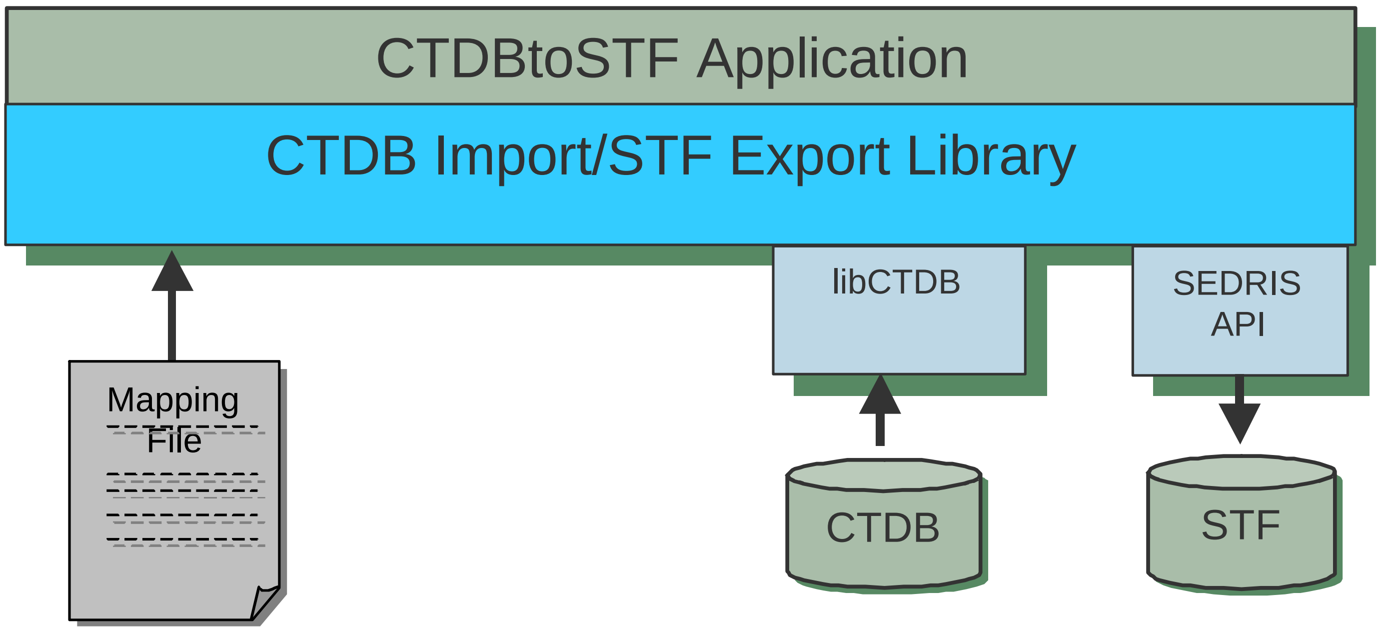
CTDB Format: MES

- A CTDB MES is a 3D Structure that has interiors and apertures. The interiors are used to represent the complete structure, while the apertures are used to provide entrance and exit portals.
- Type of MES models include:
  - Bridges
  - Tunnels
  - Buildings
  - Caves
- Complex Data Structures
  - Geometry information
  - Topology information
  - Interior and Exterior representations

CTDBtoSTF Agenda

- Introduction to the CTDB format
- CTDB to STF
  - Software Architecture
  - Data Mappings (CTDB Data Structures to DRM Classes)
  - Geometry/Feature Classification
  - Geometry Coloring
  - GCS
  - Performance
  - Real-World Use
  - Limitations
  - Software Availability
- STF to CTDB
- Questions

Software Architecture



Software Architecture (cont.)

- CTDBtoSTF is a command line program that is linked against the CTDB Import/STF Export Library.
- The command line takes in a CTDB database, a list of options for what should be consumed, and a filename for the STF that will be exported.
- Example:

  CTDBtoSTF /GRID /TIN /LAID_LINEAR /MES my_ctdb.c7l
  my_sedris.stf

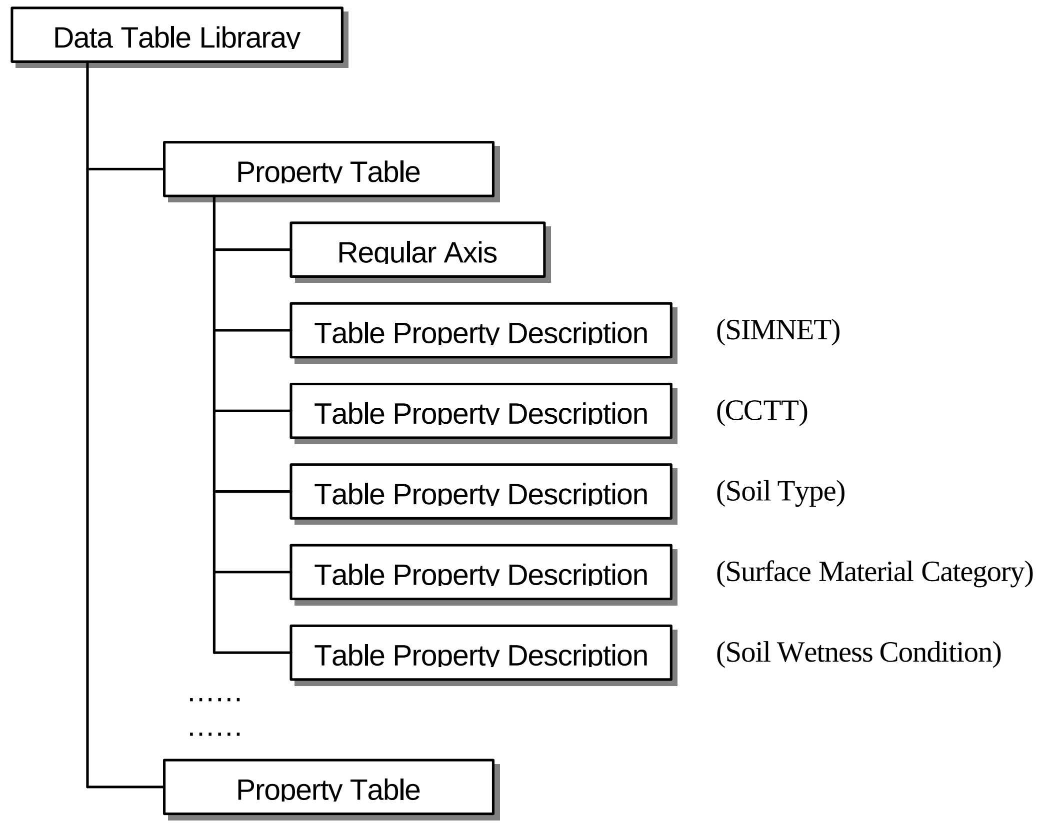
- A GUI may be provided in the future.

Data Mappings

- PAT (Polygon Attribute Table)
- Elevation and Soil Grid
- Geometry/Feature Organization
- Tree, Tree line, Laid Linear and Volume Model
- Derived Geometry
- TIN, MicroTerrain and Canopies
- Abstract Features
- Topology

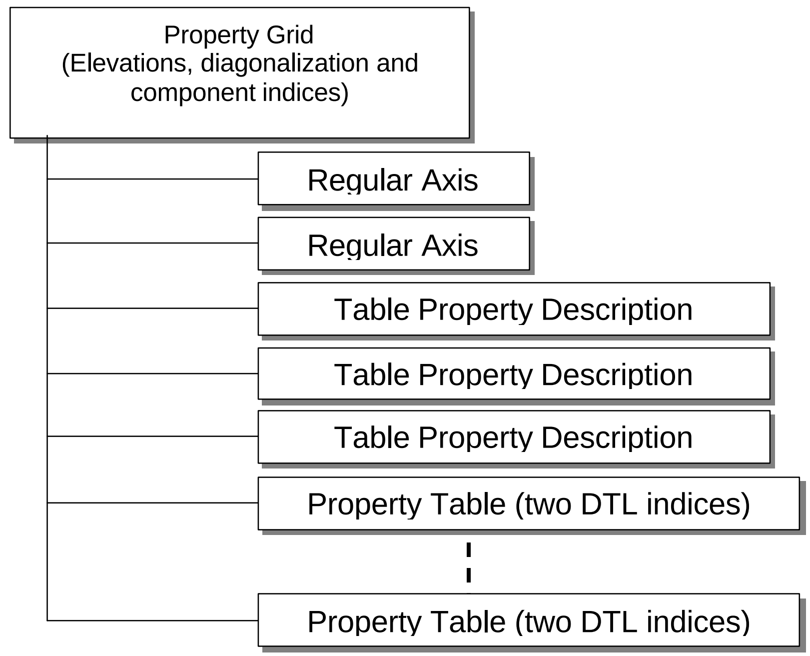
Data Mappings: PAT (Polygon Attribute Table)

- The CTDB PAT is stored in a SEDRIS <Data Table Library>.
- Each CTDB PAT entry is stored in a separate SEDRIS <Property Table>.
- CTDBtoSTF supports any number of FACC columns in the PAT.
- The graph to the right shows the default FACC attributes:
  - Soil Type
  - Surface Material Category
  - Soil Wetness Condition



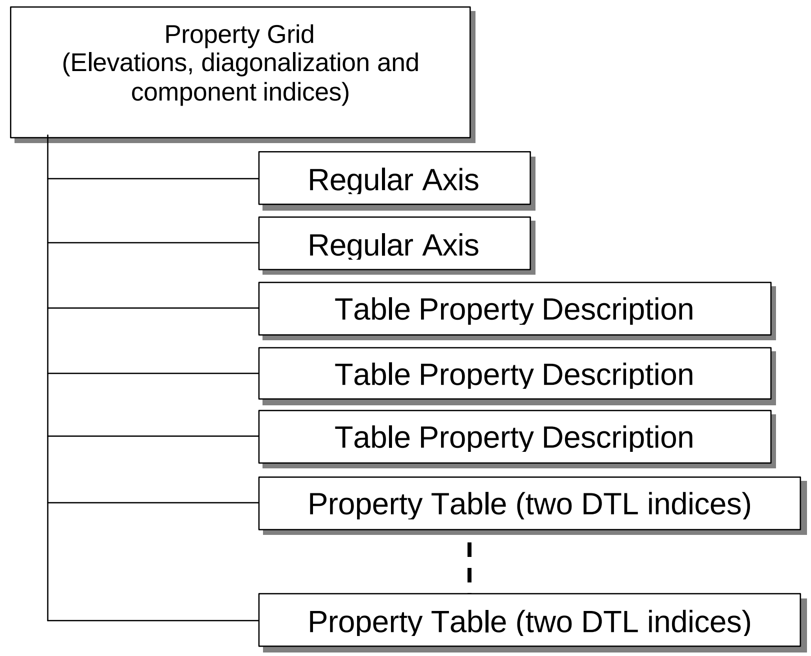
Data Mappings: Elevation Grid and Soil Grid

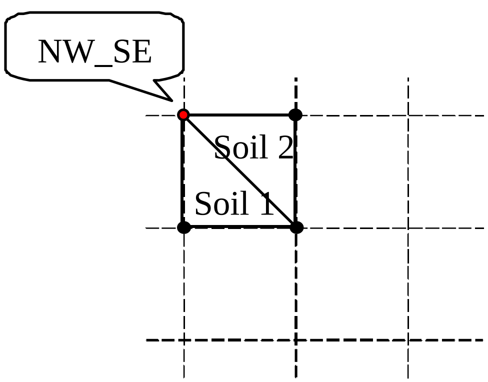
- Elevation and Soil Grids are stored in a single SEDRIS <Property Grid>.
- The <Property Grid> has three element types.
  - EAC_ELEVATION_OF_SURFACE
  - EAC_GRID_DIAGONALIZATION
  - EAC_INDEX_TO_COMPONENT



Data Mappings: Elevation Grid and Soil Grid (cont.)

- The CTDB Soil Grid stores two indices for each post (each index corresponds to one triangle in the quad).
- To mimic this, each EAC_INDEX_TO_COMPONENT Attribute Enumerant in the <Property Grid> refers to a unique <Property Table> component that stores two indices into the Data Table Library.
- These two indices correspond to the appropriate PAT entries.





Data Mappings: Elevation Grid and Soil Grid (cont.)

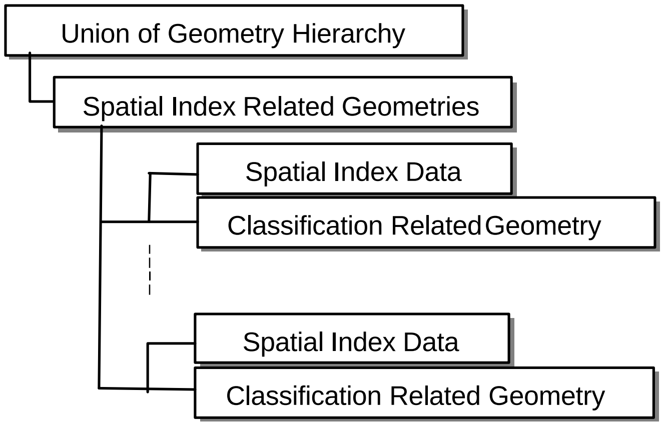
- An option allows CTDBtoSTF to convert CTDB elevation data into SEDRIS <Polygons>.
- The conversion is on patch basis and the converted<Polygons> are stored under the proper patch.
- <Polygons> of the same soil type will be grouped together, and that group will store, as a component, the <Property Table> corresponding to the appropriate PAT entry.
- CTDBtoSTF will not build <Polygons> from the CTDB Elevation/Soil Grid for patches that are flagged as being TINed.

Data Mappings: Geometry / Feature Organization

- CTDB stores Physical Features on a patch-basis.
- This patch organization is maintained in the generated STF, using <Spatial Index Related Geometry/Features> objects.
- <Spatial Index Related Geometry/Features> provides a means to organize its components into tiles.

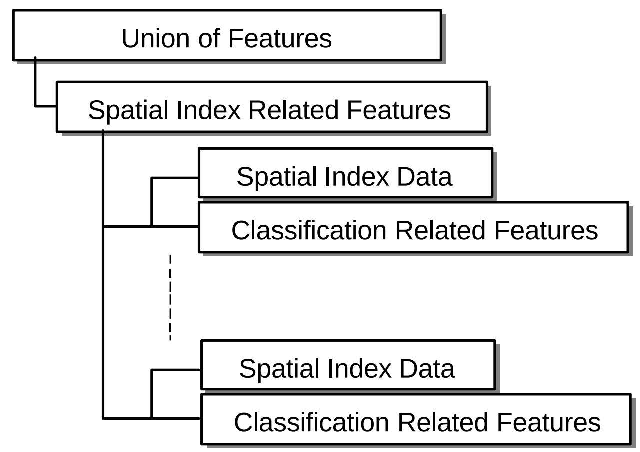
Data Mappings: Geometry Organization

- <Spatial Index Data> objects are used to store the row and column index of each tile.
- CTDBtoSTF groups features of the same type (meaning they reference the same PAT entry) together.
- The groups are then organized under a <Classification Related Geometry>.



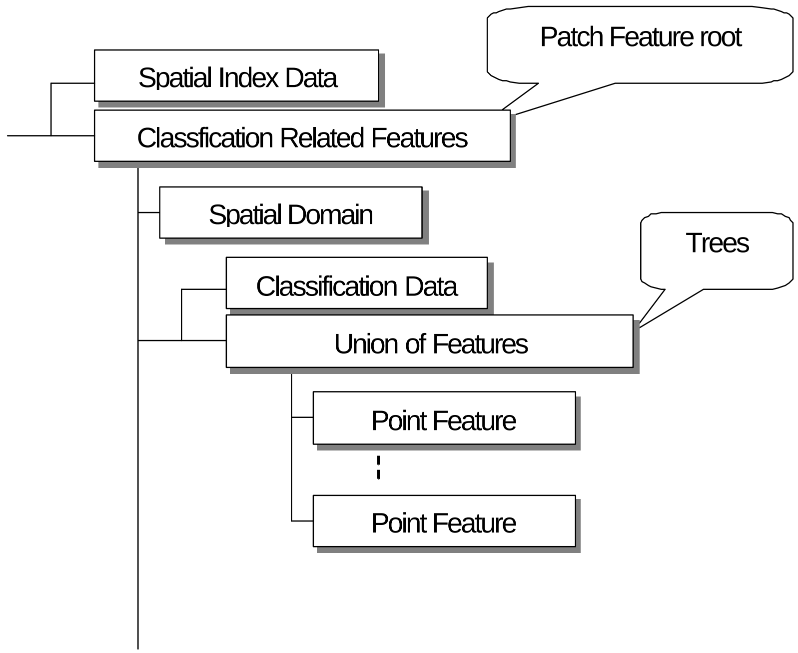
Data Mappings: Feature Organization

- The SEDRIS features produced by CTDBtoSTF are organized in an identical fashion to the geometry.



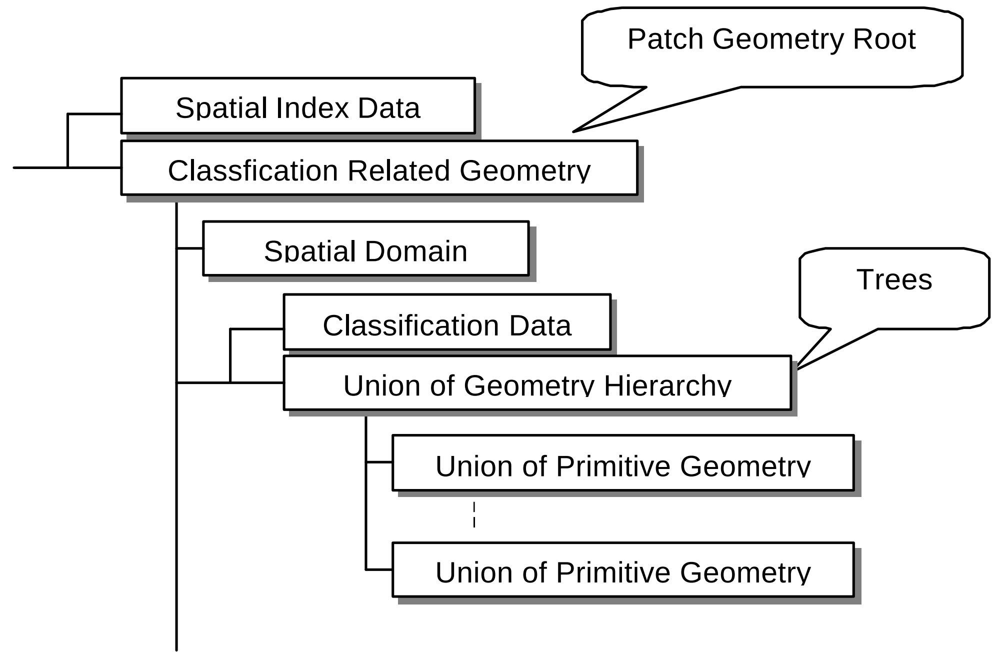
Data Mappings: Tree (Feature)

- A Tree in CTDB is stored in the generated STF as a <Point Feature>.
- A <Feature Node> is created with a 3D location storing the base of the tree.
- Additional information about the <Point Feature> are stored as <Property Values>.
  - Height
  - Radius
  - Foliage Height



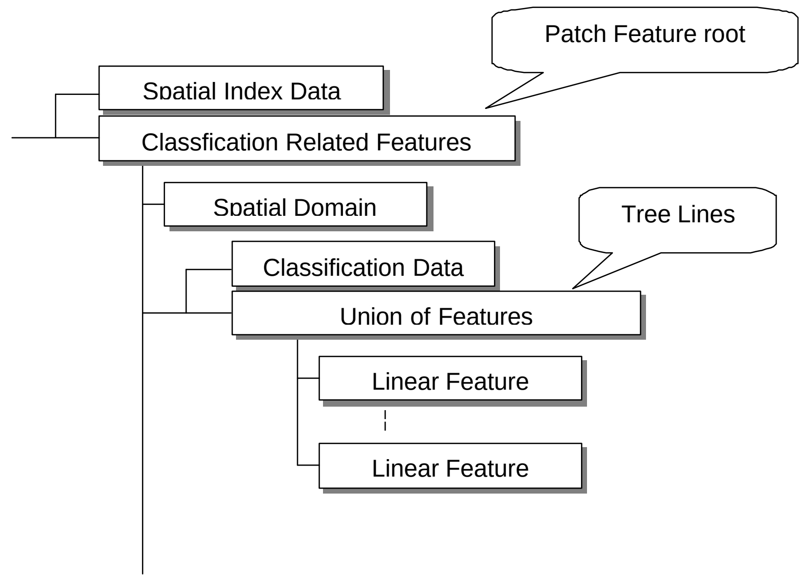
Data Mappings: Tree (Geometry)

- A <Polygon> with stamp behavior will be created for the tree as well.
- A <Union of Primitive Geometry> is creates that stores the stamp <Polygon> as a component.
- This <Union of Primitive Geometry> and the <Point Feature> are associated with each other.



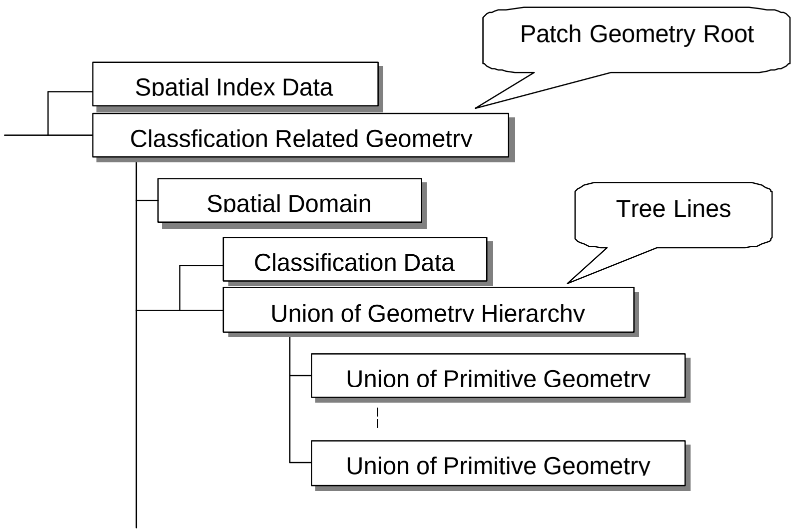
Data Mappings: Tree-Line (Feature)

- A Tree-Line in CTDB is mapped into a <Linear Feature>.
- A <Feature Edge> is created with 3D locations representing the base of the tree.
- Additional information about the <Linear Feature> is stored in <Property Values>.
  - Height
  - Radius
  - Foliage Height



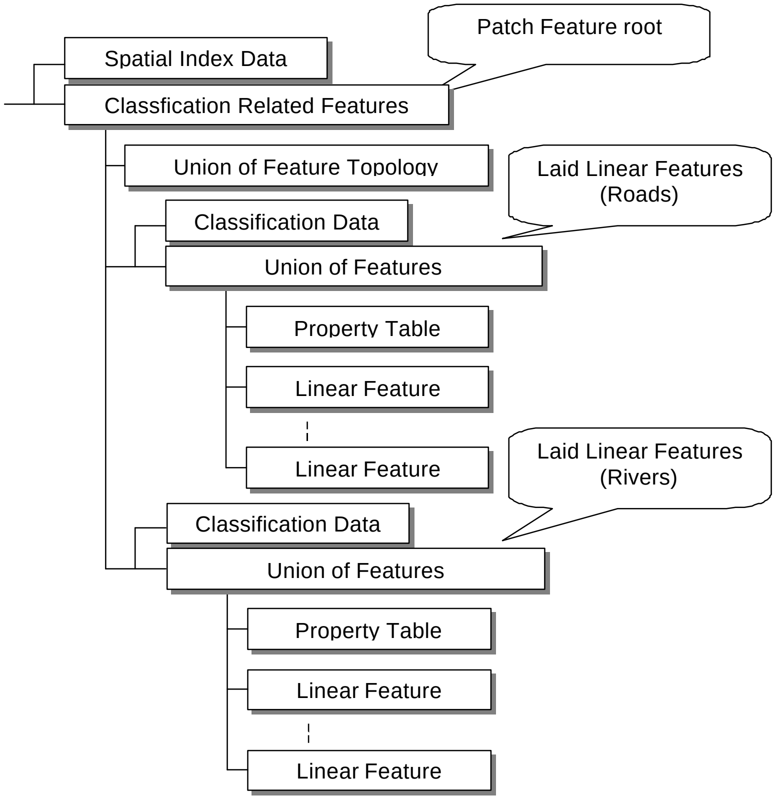
Data Mappings: Tree-Line (Geometry)

- A <Union of Primitive Geometry> will be created for the tree-line.
- It contains vertical polygons that are derived from two adjacent vertices on the tree line and the terrain heights at those locations.
- This <Union of Primitive Geometry> and the <Linear Feature> are associated with each other.



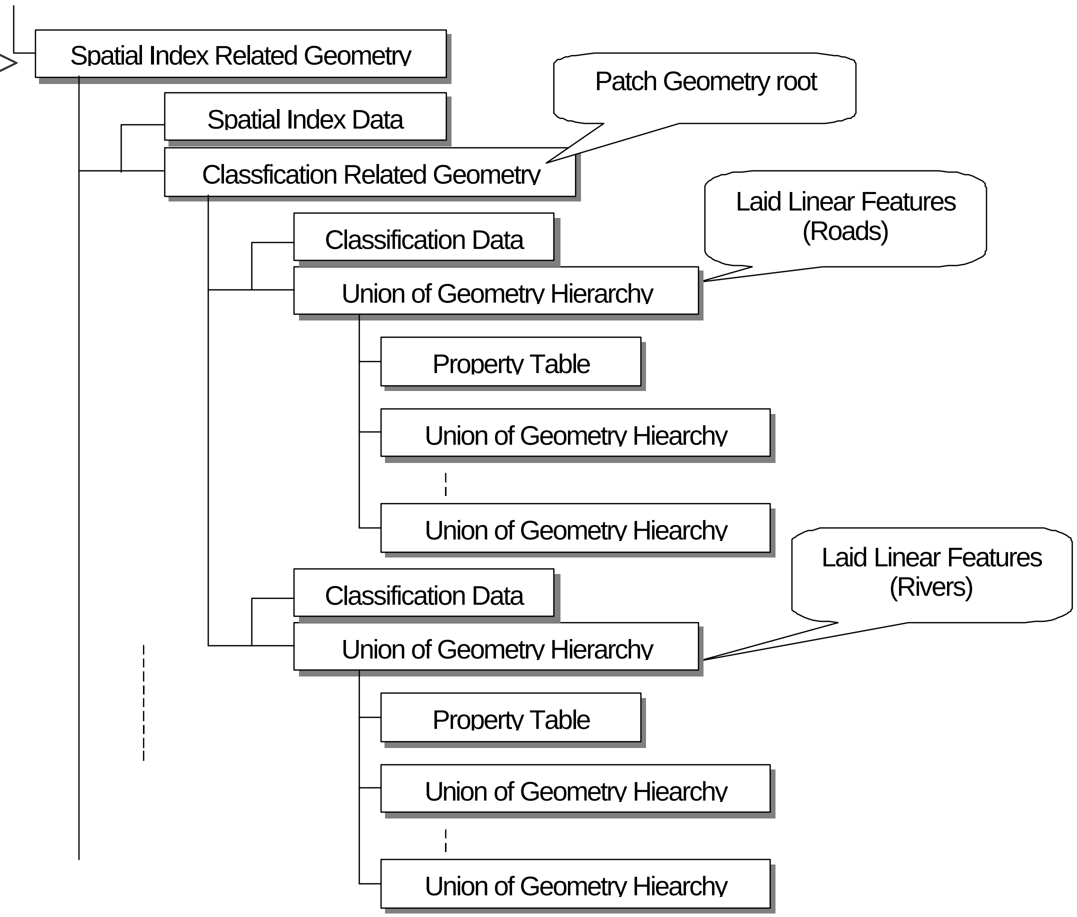
Data Mappings: Laid Linear (Feature)

- CTDB Laid Linear features are roads and rivers.
- They are stored in the generated STF as a <Linear Features>.
- This <Linear Feature> will have a <Property Value> storing the width of road or river.



Data Mappings: Laid Linear (Geometry)

- A <Union of Geometry Hierarchy> will be created for each Laid Linear.
- It contains polygons (rectangles) that are derived from two adjacent vertices on the Laid Linear.
- The <Union of Geometry Hierarchy> and the <Linear Feature> are associated with each other.



Data Mappings: Volume Model (Feature)

- CTDB Volume Models contain only roof-line vertex information.
- This roof-line is converted into an <Areal Feature>.
- Each edge of the volume contains a <Property Value> to indicate the height above terrain, unless the height is uniform, in which case one <Property Value> is used.

Data Mappings: Volume Model (Geometry)

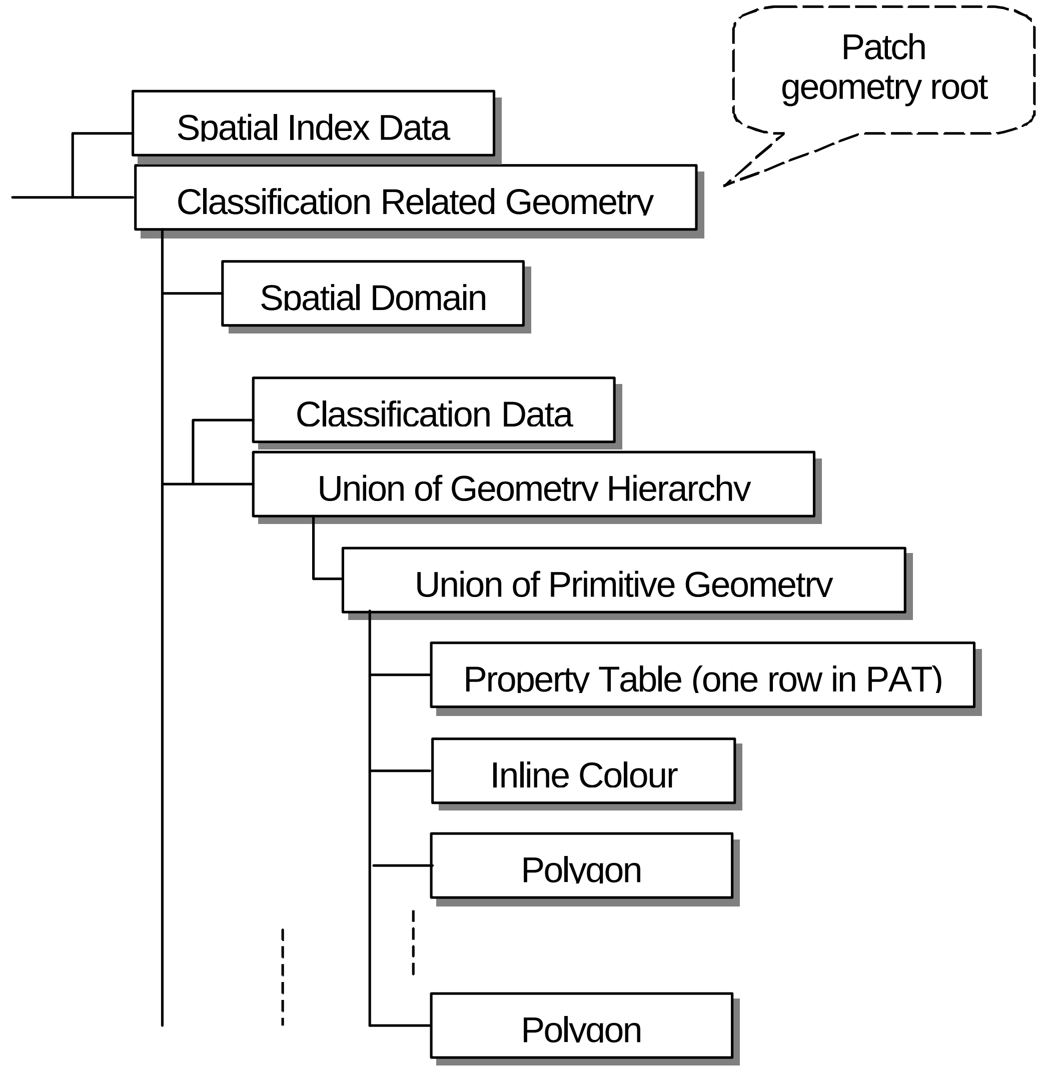
- A <Union of Primitive Geometry> is created as the derived geometry for volume models.
- Only the vertical polygons are created for walls in the volume models.
- The <Union of Primitive Geometry> and the <Areal Feature> are associated with each other.
- In the future, CTDBtoSTF will derive the roof polygons as well.

Data Mappings: Derived Geometry

- The geometric representations of the CTDB Elevation Grid, Trees, Tree-Lines, Laid Linears and Volume Models are derived in CTDBtoSTF.
- These polygonal representations are not stored directly in the CTDB.

Data Mappings: TIN

- TINs in CTDB are converted into SEDRIS <Polygons>.
- Polygons that have the same attributes are grouped into one <Union of Primitive Geometry>.
- A <Property Table>, which is one of the PAT entries, is added under this <Union of Primitive Geometry> to store the attributes of these polygons.



Data Mappings: MicroTerrain

- In CTDB, there are four types of MicroTerrain:
  - Base (Standard Microterrain)
  - Multi-Elevation
  - Default (only used if there isn't Base or Multi-Elevation MicroTerrain)
  - Water
- Currently, CTDB to STF is only supporting Base MicroTerrain.
- MicroTerrain polygons are tagged appropriately and stored in the geometry tiles as SEDRIS <Polygons> in the same manner as the TINs.

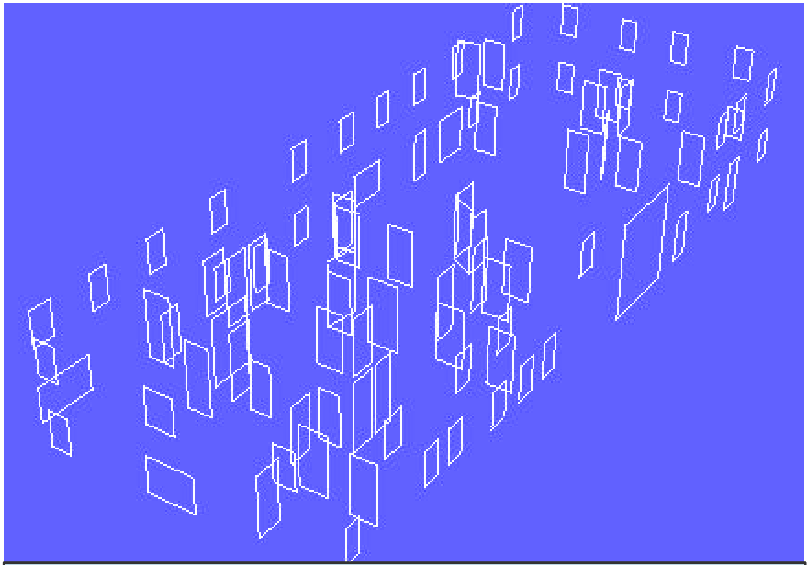
Data Mappings: Canopy

- Canopies are stored as <Polygon> objects.
- Canopies are tagged appropriately and stored in the geometry tile as SEDRIS <Polygon> objects.
- Canopies in CTDB store a flag to indicate whether a point lies on the canopy edge. This flag is currently not being retained.

NOTE: CTDB also stores canopies as a footprint with the abstract features, which are translated into SEDRIS <Areal Features>.

Data Mappings: MES (Multi-Elevation Structures)

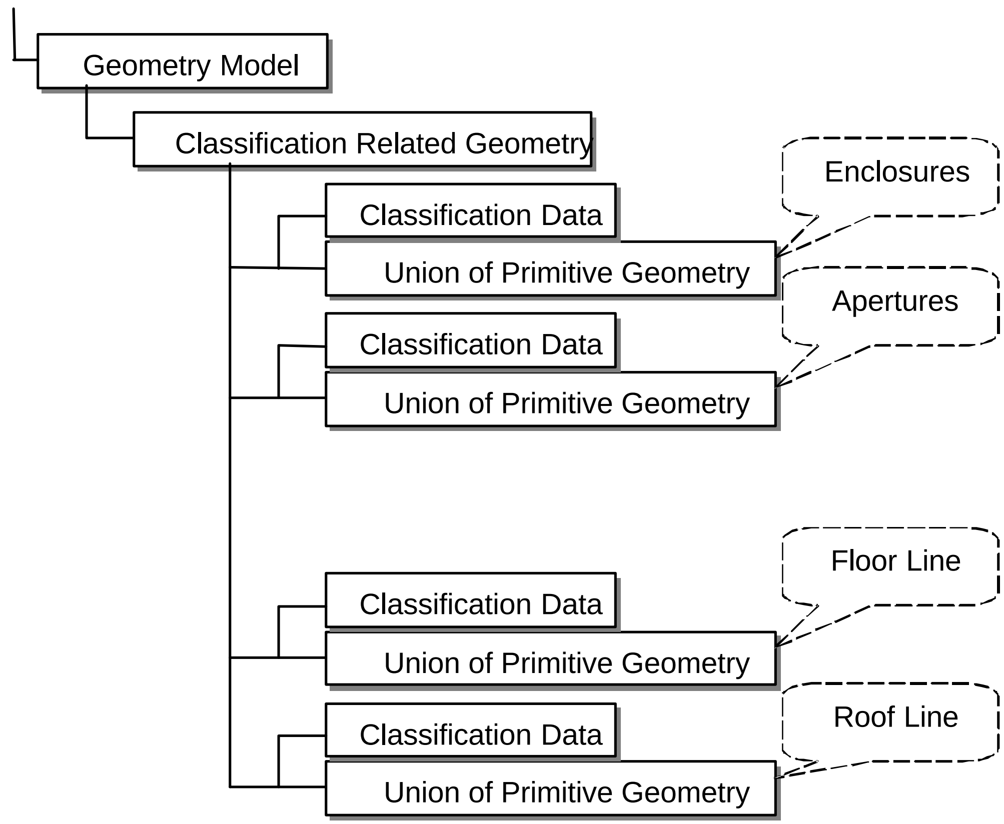
- Unlike standard CTDB volumes which only contain roofline vertices, MES volumes contain complete structure information.
  - Roofline and Ground line
  - Floor lines
  - Enclosures
  - Apertures
- Each of these portions is classified using a <Classification Related Geometry>.
- The MES topological information is not currently being retained.



Side-by-Side Screen Shot Displaying all the MES' apertures

Data Mappings: MES (Multi-Elevation Structures)

- The roof-line and ground-line are stored in world coordinates in the CTDB. The enclosure, aperture and floor-line polygons are stored in local space and hence can be indexed.
- CTDBtoSTF translates the vertices for the roof-line and ground-line and stores them with the Enclosures and Apertures in a <Geometry Model>.
- The <Geometry Model> is then instanced in the environment with the proper world transformation values.



Data Mappings: Abstract Features

- Abstract Features in CTDB are stored in a quad tree, as opposed to the patch organization of the Physical Features.
- This quad tree structure is not retained in the STF.
- Instead, all of the Abstract Features for a CTDB are grouped into one <Union of Features>.
- The quad tree structure will be retained in the future.

Data Mappings: Abstract Features (cont.)

- The CTDB Abstract Feature type is used to determine the appropriate SEDRIS Class.
- A few examples are shown below.

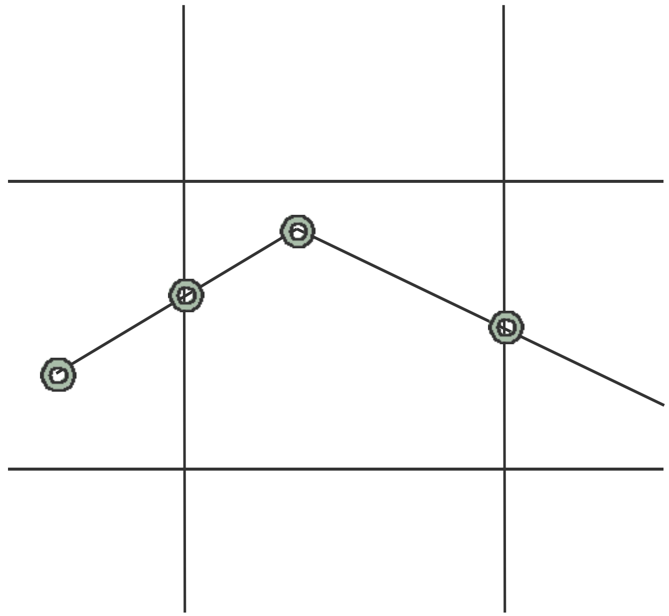
CTDB	SEDRIS Class
Soil Defragmentation Area	<Areal Feature>
Canopy Footprint	<Areal Feature>
Railroad	<Linear Feature>

Data Mappings: Topology

- In CTDB, the topology of Abstract and Physical Linear Features are made up by edges and nodes.
  - Edge: *An edge is a sequence of Physical or Abstract Linear Features (sub-edges) that traverse the space between nodes*.
  - Node: *A node is the location where 1 or more edges intersect.*
- CTDBtoSTF converts:
  - *CTDB Edge => <Linear Feature>*
  - *CTDB Sub-edge => <Feature Edge>*
  - *CTDB Node => <Feature Node>*

Data Mappings: Topology

- Cross Boundary Problem:
  - CTDB edge (thus a SEDRIS <Linear Feature>) may cross patch boundaries.
  - CTDBtoSTF solution: “Hard” boundary for <Feature Edges>, “Soft” boundary for <Linear Features>.
  - Therefore, multiple tiles may share the same <Linear Feature>.



- Each circle represents a CTDB Node.
- Each edge between points is a CTDB Sub-Edge.
- The entire line is a CTDB Edge.

Data Mappings: Topology (cont.)

- CTDB stores a flag with each Node indicating whether a bridge exists at that point.
- When that flag is set, CTDBtoSTF will build a <Point Feature> to represent that bridge.

Geometry / Feature Classification

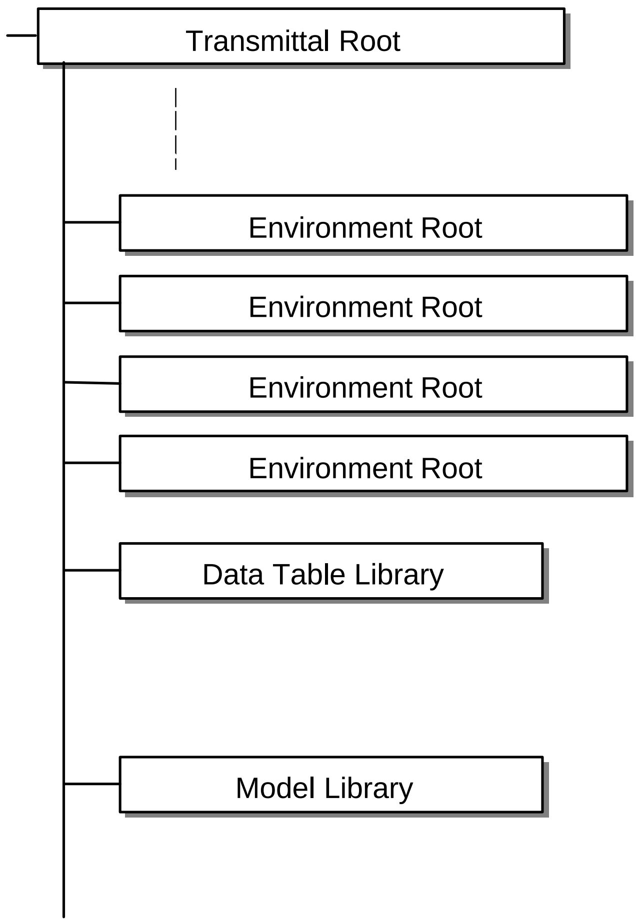
- Features that are of the same type are stored together and tagged with an appropriate EDCS Classification Code (ECC).
- The ECC is determined by the CTDB feature and soil type.
  - For example, a laid linear with a soil type of water would be translated into ECC_RIVER_OR_STREAM.
  - A laid linear with a soil type of road would be translated into ECC_ROAD.
  - A soil defrag, a type of abstract feature, with a soil type of water would be classified as ECC_LAKE.
- The mappings are stored in an ASCII file and can be edited for specific needs.

Geometry Coloring

- All geometries in the generated STF are assigned a color.
- The color is determined by the CTDB feature and soil type.
  - For example, a laid linear with a soil type of water will have a blue color.
- The colors are specified in the mapping file, which is editable.

GCS (Global Coordinate System)

- CTDB to STF consumes multi-cell GCS STFs.
- Each cell is stored in a separate <Environment Root>.
- PAT entries from each CTDB cell are stored under the <Data Table Library> sequentially.
- GCS Support is still in development.



GCS (cont.)

- CTDB Format 7 supports an older version of GCS, as well as the newer model, GTRS.
- The SEDRIS SRM only uses the newer version of GCS.
- The SEDRIS SRM provides functions to convert from the older style to the new version.
- CTDBtoSTF will use these routines.

Performance

Database Name	CTDB Size	STF Size	STF Size (with derived Geometry)	Processing Time (with derived Geometry)
Bellevue	84.5 KB	60 KB	429 KB	4 seconds
McKenna	583 KB	470 KB	563 KB	6 seconds
Philly	2.39 MB	3.8 MB	7.55 MB	2.9 minutes
NTC	2.17 MB	2.2 MB	13.7 MB	3.3 minutes
Hunter	5.24 MB	5.1 MB	19.2 MB	4 minutes
Bosnia	9.61 MB	13.4 MB	57.1 MB	9 minutes

Windows 2000

PIII 700, with 256 Megabytes of RAM

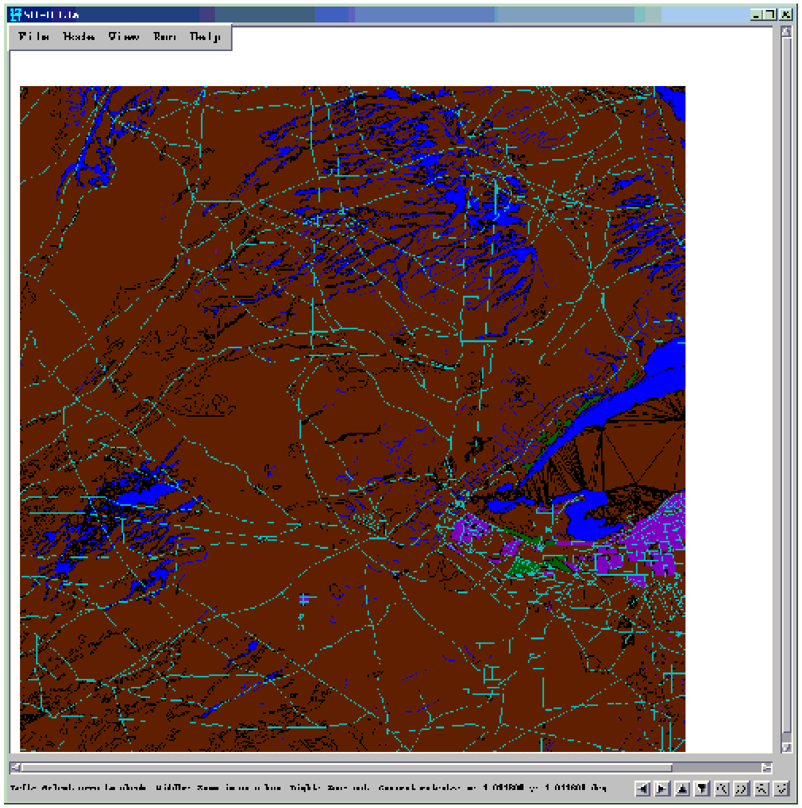
SEDRIS 3.1

Real-World Use: STFtoCTDB

- By using STFtoCTDB in conjunction with the CTDBtoSTF, one can have a round trip conversion.
- This makes it possible to use SEDRIS editing tools, such as Focus and SEE-IT, to indirectly edit CTDBs.
  - Convert CTDB to STF.
  - Modify resulting STF with SEDRIS tools.
  - Convert the modified STF back into CTDB.
  - Continue CTDB usage.

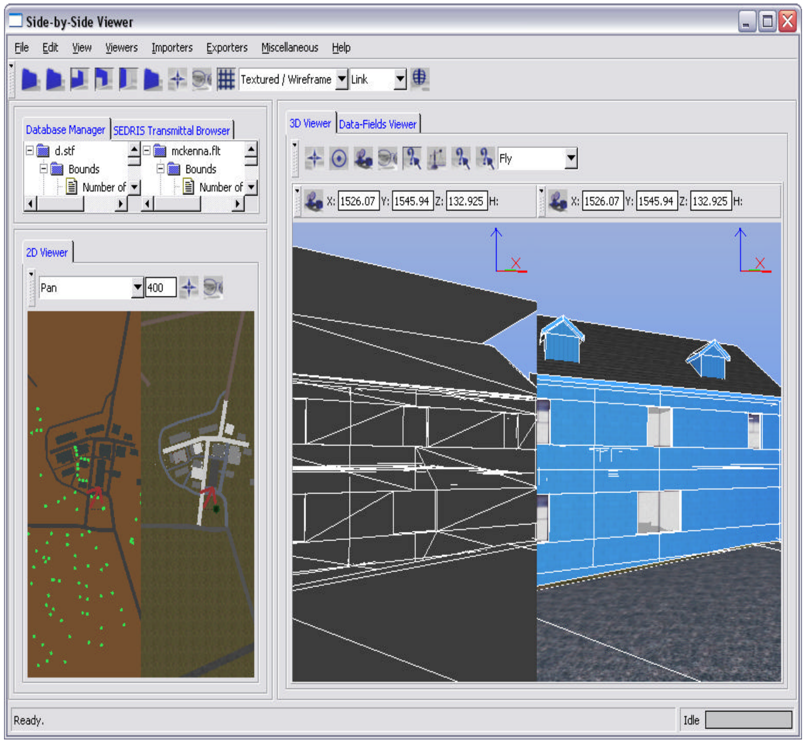
Real-World Usage: SEE-IT

- SEE-IT can evaluate terrain databases to locate conditions (e.g., road and river intersections that do not have bridges).
- By using CTDBtoSTF you can use SEE-IT to, indirectly, evaluate the CTDB.



Real-World Usage: Side-by-Side

- Side-by-Side, developed by AcuSoft, provides correlation, conversion and intensification features.
- The screen-shot shows a comparison between an STF (created from CTDB) with the OpenFlight version of the database.



CTDBtoSTF Limitations

- Abstract Features Organization: the quad tree structure is not maintained in the generated transmittals.
- Not all forms of MicroTerrain are being stored.
- Only supports CTDB Format 7.
- GCS support is not yet complete.

Software Availability

- CTDBtoSTF is still in the process of beta testing.
- The software is currently available to SEDRIS associates.
- Visit *http://tools.sedris.org* for the latest information about the software.

STF to CTDB Agenda

- Introduction to the CTDB format
- CTDB to STF
- STF to CTDB
  - Mapping STF Data to CTDB
    - Understanding the DRM and EDCS to map data
    - Effective Handling of Multiple Representations
    - STF to CTDB Data Mappings
  - Overview of Operation
  - Tool Status
  - Real World Usage
- Questions

Mapping STF Data - DRM

- Compiler understands certain primitive DRM Classes
  - Geometry
    - <Polygons>
      - Terrain, Canopy Roofs
    - <Geometry Models>
      - Buildings
    - <Property Grids>
      - Terrain Elevation
  - Features
    - <Point Features>
      - Trees, Buildings, Bridges, etc.
    - <Linear Features>
      - Roads, Rivers, Bridges, etc.
    - <Areal Features>
      - Soil defragmentation areas, canopy footprints, etc.
- Compiler understands certain organizational DRM Classes
  - <Level of Detail Related>
    - Can chose HIGH or LOW level of detail
  - <Spatial Index Related>
    - Used to cut out smaller CTDBs efficiently

Mapping STF Data - EDCS

- Compiler uses EDCS Classification and Attribution to interpret object meaning.
  - EDCS Classification found in <Classification Data> DRM object as component to primitive DRM object.
  - EDCS Attribution is found in <Property Value> DRM object as component to primitive DRM object.
    - Can be:
      - Floats, Integers, Booleans, EDCS Enumerants, or Intervals

Mapping STF Data – Putting it Together

- DRM Primitives combined with EDCS Classification and EDCS Attribution represent a specific object with real properties.
  - <Linear Feature>
    - <Classification Data> ECC_ROAD
    - <Property Value> EAC_WIDTH of 10 meters
  - <Polygon>
    - <Classification Data> ECC_TERRAIN
    - <Property Value> EAC_SURFACE_MATERIAL_TYPE of SAND

STF to CTDB Data Mapping Topics

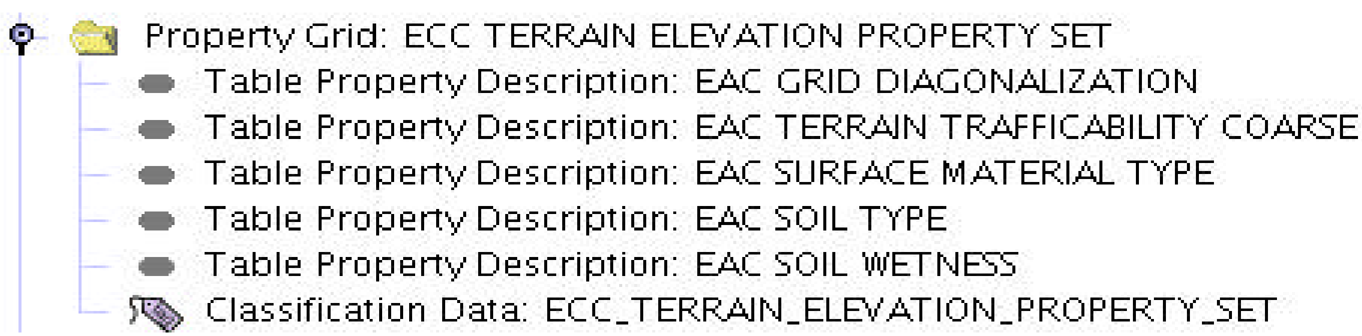
- General Requirements
  - SRF Requirements
- Requirements for CTDB Terrain Data
  - Gridded Terrain Elevation Data
  - Polygonal Terrain Elevation Data
  - Terrain Soil / Mobility information
- Requirements for CTDB Feature Data
  - Buildings and Obstacles
  - Roads and Rivers
  - Bridges
  - Trees and Tree lines
  - Canopies
  - Soil Defrags
  - Railroads, Pipelines, Power lines

Data Mappings: General Requirements

- SRF Requirements
  - The following SEDRIS SRM SRFs are currently supported:
    - SRM_SRF_3D_GD (Geodetic)
    - SRM_SRF_3D_AUTM (Augmented Universal Tranverse Mercator)
    - SRM_SRF_3D_ATM (Augmented Tranverse Mercator)
    - Support of additional SRFs is possible, if needed
      - But only if SRM supports conversion from that SRF to the AUTM SRF
  - Other issues:
    - CTDB extents may not cross UTM zone boundaries

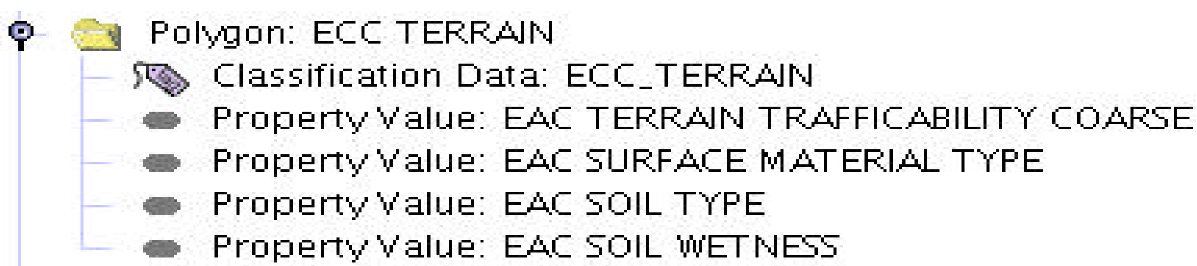
Data Mappings: Gridded Terrain Data

- Used to fill CTDB elevation grid posts and corresponding soil grid.
  - Can also be used to create CTDB TIN data.
- Contained in a <Property Grid>
  - SRFs supported are the same listed on previous slide.
  - Required <Table Property Descriptions>
    - EAC_TERRAIN_ELEVATION
    - Trafficabilty Information
      - EAC_TERRAIN_TRAFFICABILITY_COARSE (SIMNET) or
      - EAC_TERRAIN_TRAFFICABILITY_MEDIUM (CCTT)
    - EAC_GRID_DIAGONALIZATION
    - EAC_SOIL_TYPE
    - EAC_SOIL_WETNESS
    - EAC_SURFACE_MATERIAL_TYPE
- STF Structure



Data Mappings: Polygonal Terrain Data

- Used to create CTDB TIN and microterrain
  - Microterrain only created where elevation grid is valid, otherwise TIN is used.
- Stored as <Polygon> objects.
  - Required Attribution in <Property Value> objects
    - Trafficability
      - EAC_TERRAIN_TRAFFICABILITY_COARSE (SIMNET) or
      - EAC_TERRAIN_TRAFFICABILITY_MEDIUM (CCTT)
    - EAC_SOIL_TYPE
    - EAC_SOIL_WETNESS
    - EAC_SURFACE_MATERIAL_TYPE
  - Attribution used to create entries in CTDB Polygon Attribute Table
- Can also be used to create CTDB gridded elevation data.
- STF Structure:



Data Mappings: Buildings and Obstacles

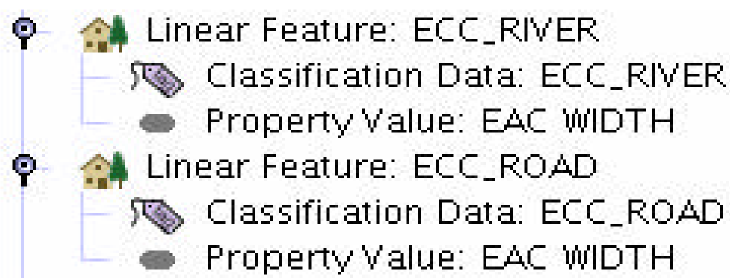
- Used to create CTDB volume models
- Stored as <Point Features>, <Areal Features> or <Geometry Models>
  - <Point Feature> Requirements
    - Point represents exact center of volume model roofline, and the actual roofline is derived from the given attribution.
    - Required Attribution:
      - Length, Width, Height, and Angle of Orientation
  - <Areal Feature> Requirements
    - Each point represents one point in the volume model roofline.
    - Required Attribution
      - Height
- STF Structure:

Data Mappings: Buildings and Obstacles

- Special Case: <Geometry Models>
  - No attribution required
  - Building Algorithm:
    - Computes a roofline by finding all vertical or near vertical polygons (within ±15 degrees) and attempting to find a path that traverses along them to create a roofline.
  - Obstacle Algorithm:
    - Computes a simple bounding box by finding the x,y, and z extents, and encodes the top of the bounding box as the roofline. Useful for small obstacles (fire hydrants, signs, vehicles) and unusually shaped buildings that fail the building algorithm.
  - When <Geometry Model> is associated to a <Point Feature> or <Areal Feature>, the feature is considered an alternate representation and processed instead.

Data Mappings: Roads and Rivers

- Stored as <Linear Features>
  - Required Attribution
    - Width
- Roads and Rivers are both stored in CTDB as Laid Linear segments, and all topology information is created by the backend compiler.
  - If specified via a command line argument, implicit bridges can be created at Road and River intersections during the topology creation process.
- STF Structure:

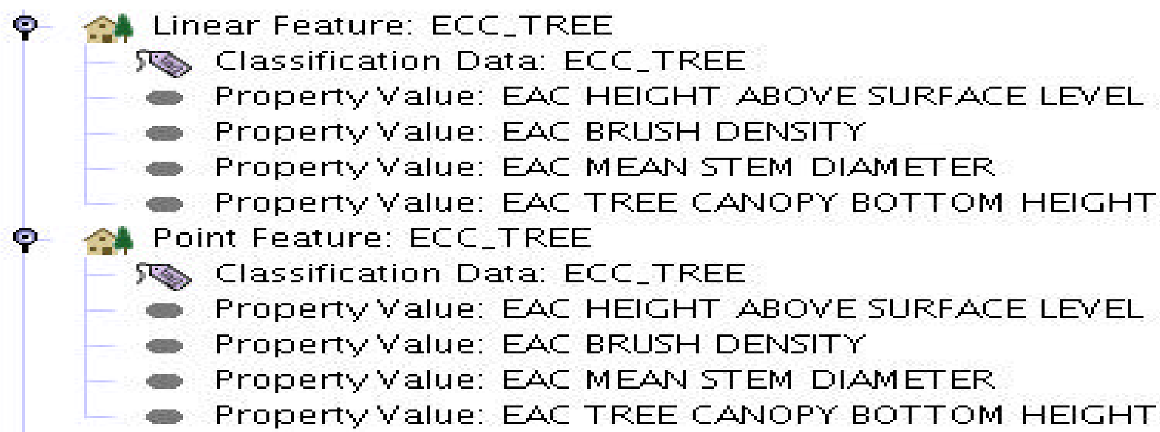


Data Mappings: Bridges

- Stored as <Linear Features>
  - Required <Property Value> Attribution
    - Length, Width, Height, Angle of Orientation
  - 3D MES Bridge structure is created using these attributes.
    - MES bridge works in DISAF 8.0
    - Does not work in any other SAF
- Users should consider mapping bridges as roads and allowing the backend to implicitly create bridges at intersections.
- STF Structure:

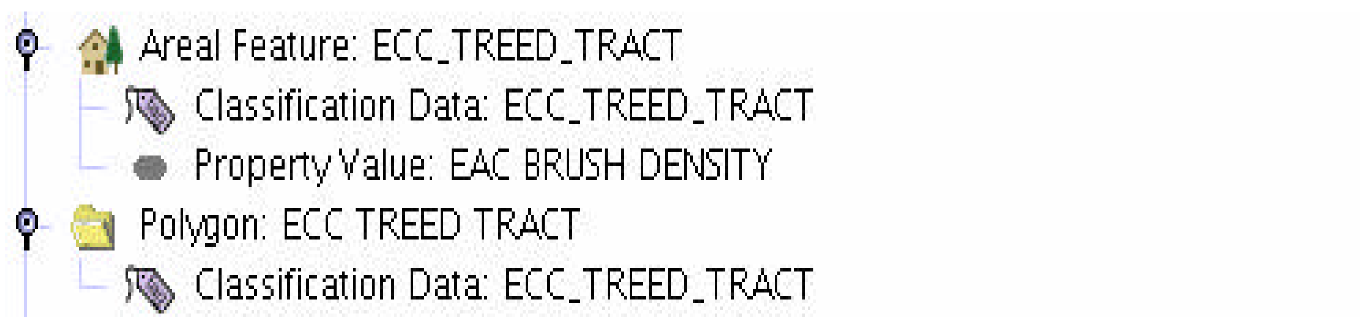
Data Mappings: Trees and Tree lines

- Stored as <Point Features> or <Linear Features>
  - Required <Property Value> Attribution
    - Height – Total Height of Tree
    - Foliage height – Height of foliage in meters measured from the top of tree.
    - Opacity – Fullness of foliage.
    - Diameter – Diameter of foliage
  - All CTDB trees are stored as Tree lines, individual <Point Feature> trees are stored as a Tree line with only one vertex.
- STF Structure:



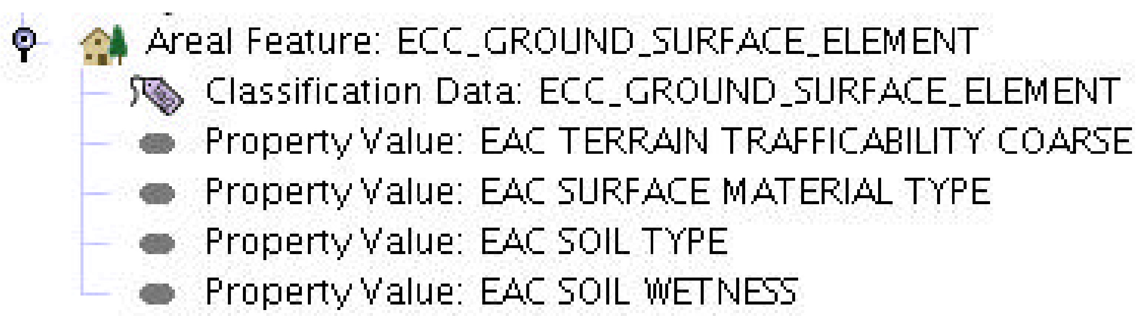
Data Mappings: Canopies

- Stored separately as both <Areal Features> and <Polygons>
  - <Areal Features> describe the boundary of the canopy, and whether or not it can be seen through and/or driven through. This information is used to create both abstract and physical feature canopy objects.
    - Required <Property Value> Attribution
      - Density – used to determine the transparency and penetrability of the canopy.
      - Transparency information stored in physical feature canopy
      - Penetrability stored in abstract feature canopy
  - <Polygons> describe the roof the the canopy.
    - Required <Property Value> Attribution
      - None
    - Stored only as physical feature canopy object.
- STF Structure:



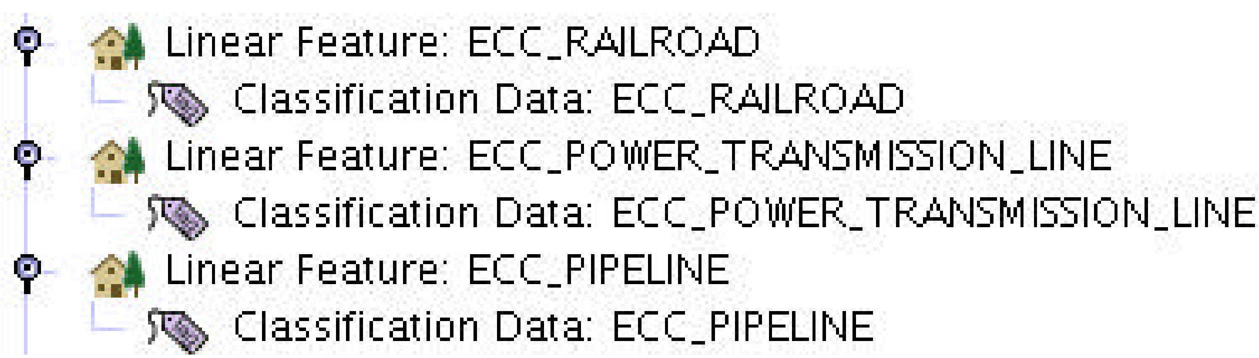
Data Mappings: Soil Defragmentation Areas

- Stored as <Areal Features>
  - Used to describe the mobility and soil information for the represented area in the CTDB.
  - Required <Property Value> Attribution
    - Some Trafficabilty Information
      - EAC_TERRAIN_TRAFFICABILITY_COARSE (SIMNET) or
      - EAC_TERRAIN_TRAFFICABILITY_MEDIUM (CCTT)
  - Desired <Property Value> Attribution
    - EAC_SOIL_TYPE
    - EAC_SOIL_WETNESS
    - EAC_SURFACE_MATERIAL_TYPE
- Special Notes:
  - Soil defrags are used by the compiler to attribute soil/mobility information to those polygons that have none specified.
- STF Structure:



Data Mappings: Railroads, Pipelines, Power lines

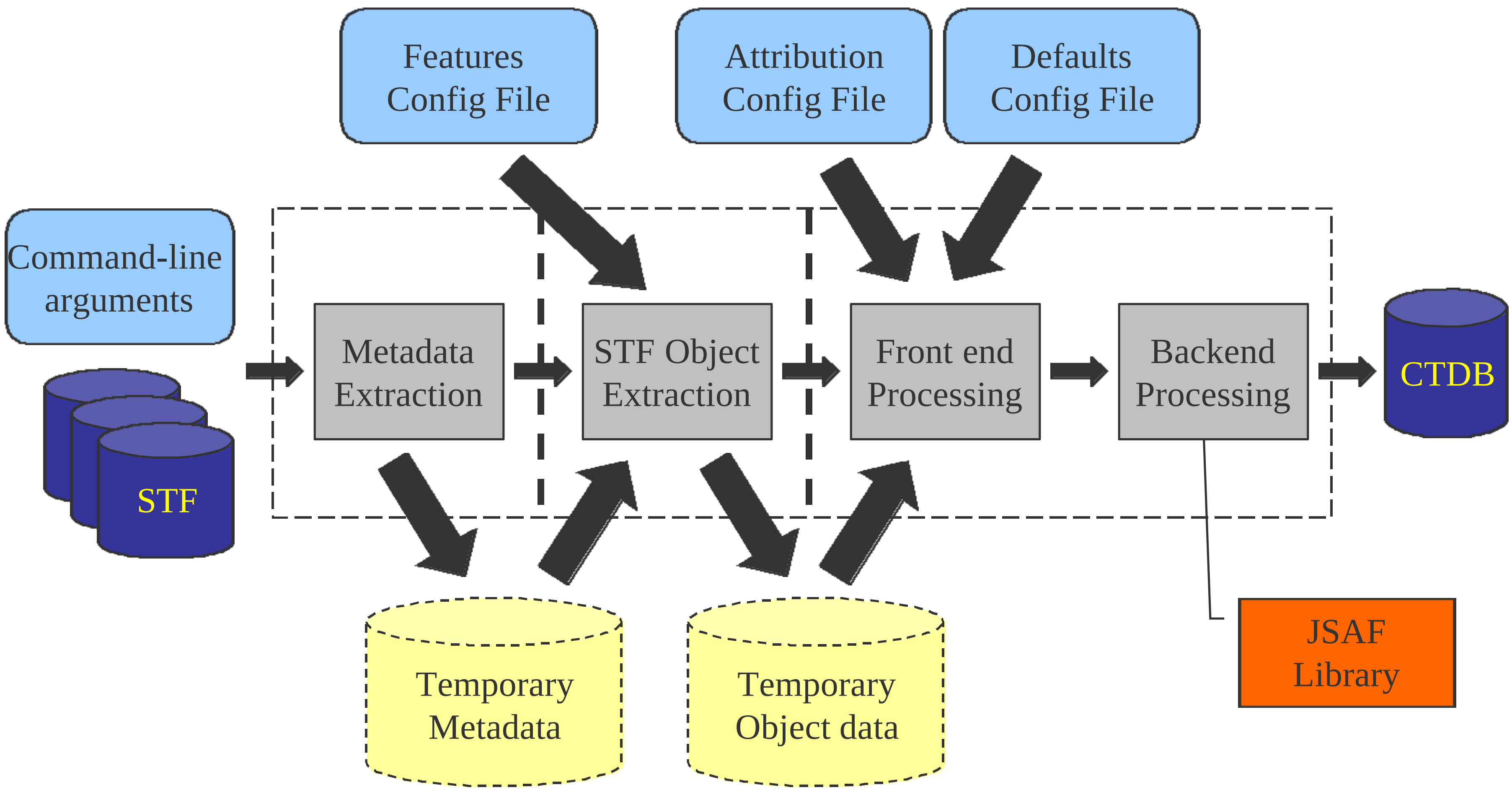
- Stored as <Linear Features>
  - Required <Property Value> Attribution
    - None
  - Stored as abstract features in CTDB.
- STF Structure:



Data Mappings: Multiple Representations

- With the power of the SEDRIS DRM, data can be stored in a variety of different representations
- In some cases, STF to CTDB can handle multiple representations
  - For buildings it can understand <Areal Features>, <Point Features> and <Geometry Models>
- However, it cannot possibly handle every representation
  - For example roads could be stored as:
    - <Linear Features> representing the route of each road with a <Property Value> to give the width.
    - <Areal Features> representing the outline of the complete road, where width would have to be reasoned from the width of the <Areal Feature> itself.
    - <Polygons> stitched into the terrain tagged as ECC_ROAD.
    - etc…
  - STF to CTDB can only handle <Linear Feature> representation
- The set of data mappings described here documents which representations the STF to CTDB does handle
  - An informal Transmittal Contents Requirement Specification for the STF to CTDB compiler.

Overview of Operation



Overview of Operation (cont)

- Configuration Files
  - Features File
    - Maps DRM and EDCS Classification Code combinations to CTDB concepts
  - Attribution File
    - Maps EDCS Attribute Codes to CTDB concepts
  - Defaults File
    - Provides ability to specify default values for attribution when none exists in the input data

Overview of Operation: Features File

- Features File Entry Example:
  - Map any <Linear Feature> with any of the three given EDCS Classification Codes to a CTDB laid linear road.
  - Map <Geometry Models> that are classified as ECC_BUILDING, <Areal Features> that are classified as ECC_ADMINISTRATION_BUILDING, and <Point Features> that are classified as ECC_BARRACK or ECC_BARN to CTDB Buildings.

```
(
  ( road
      ( SE_DRM_CLS_LINEAR_FEATURE
          ( ROAD ROAD_INTERCHANGE TRAIL CART_TRACK STREET ))
  )
  ( building
      ( SE_DRM_CLS_GEOMETRY_MODEL_INSTANCE
          ( BUILDING ))
      ( SE_DRM_CLS_AREAL_FEATURE
          ( ADMINISTRATION_BUILDING ))
      ( SE_DRM_CLS_POINT_FEATURE
          ( BARRACK BARN ))
  )
)
```

Overview of Operation: Attribution File

- Attribution File Entry Example:
  - "width" entry instructs the compiler that any <Property Value> with EAC_WIDTH or EAC_ROAD_TOTAL_USABLE_WIDTH can be interpreted as width.
  - "height" entry instructs the compiler that any <Property Value> with EAC_HEIGHT_ABOVE_SURFACE_LEVEL or EAC_PREDOMINANT_HEIGHT can be interpreted as height.
  - "density" entry instructs the compiler that any <Property Value> with EAC_BRUSH_DENSITY or EAC_WOODY_VEGETATION_DENSITY can be interpreted as density.
  - Where more than one <Property Value> is found when searching for attribution, the first one in the list that matches will be used.

```
(( width ( WIDTH ROAD_TOTAL_USABLE_WIDTH ))
 ( height ( HEIGHT_ABOVE_SURFACE_LEVEL PREDOMINANT_HEIGHT ))
 ( density ( BRUSH_DENSITY WOODY_VEGETATION_DENSITY ))
)
```

Overview of Operation: Defaults File

- Defaults File Entry Example:
  - When searching for “height” attribution of building, if no <Property Value> is found for “height” as described in the previous slide, then the default value specified here as “10” will be used.
  - Likewise for trees and tree lines, if no “foliage height” attribution is found, the default value of 8 specified in this file will be used.

```
(
  ( building
    ( height 10 )
    ( width 10 )
    ( length 10 )
    ( angle 0 ))
 ( treeline
    ( height 10 )
    ( foliage_height 8 )
    ( diameter 2 )
    ( opacity .60 ))
)
```

Tool Status: Topics

- Release Information & Schedule
- Testing
- Limitations
- Performance
- Future Enhancements

Tool Status: Release Info

- Beta Release
  - Beta releases available to SEDRIS associates and on request to non-associates. Email help@sedris.org to request beta software.
  - Last 3.0 based beta was on 6/25/02
  - First 3.1 based beta release will be shortly after STC
- Release 1.0
  - Based on SEDRIS 3.1 baseline
  - Scheduled for the end of this year
  - Will Contain
    - GCS Support
    - Limited MES support
  - Rigorous testing before release

Tool Status: Testing

- Testing Strategy
  - How to test for total correlation?
    - Coordinate systems differ.
    - Storage techniques differ.
    - Begin with elevation correlation and move on to features.
  - CTDB to STF loop back testing.
    - Goal: CTDB created from an STF that was created by the CTDB to STF tool should be identical to the original CTDB.
    - Likewise with STF loop backs.

Tool Status: Limitations

- Current Limitations:
  - No GCS Support
  - No MES Support
  - Only supports CTDB format 7

Tool Status: Performance

Statistics for Bellevue STF	
Database Size (Meters)	10000x10000
Database Type	Rural
3.1 Input STF Size (MB)	3.8
3.1 Output CTDB Size (MB)	0.6
3.1 Total Running Time (minutes)	00:18.9
3.1 Temp File Size (MB)	4.2

Statistics for VERTS Philly STF	
Database Size (Meters)	2000x1000
Database Type	Urban
3.1 Input STF Size (MB)	15.7
3.1 Output CTDB Size (MB)	2
3.1 Total Running Time (minutes)	01:35.9
3.1 Temp File Size (MB)	50.7

* All statistics collected on Dual 1.7 GHz P4 Xeon with 768 MB of physical RAM running Linux Red Hat 7.2

Tool Status: Performance

Statistics for DTED Level 0	
Database Size (Meters)	90500x110000
Database Type	Terrain Elevation
3.1 Input STF Size (MB)	16.7*
3.1 Output CTDB Size (MB)	8.9
3.1 Total Running Time (minutes)	0:30.17
3.1 Temp File Size (MB)	6.87
* STF contains Level 0, Level 1, and Level 2 DTED data	

- Notes
  - STF contains information that cannot be stored in resulting CTDB
    - Information is not converted
  - Running time not solely a result of database size but is also affected by database content
    - Processing geometry models takes longer

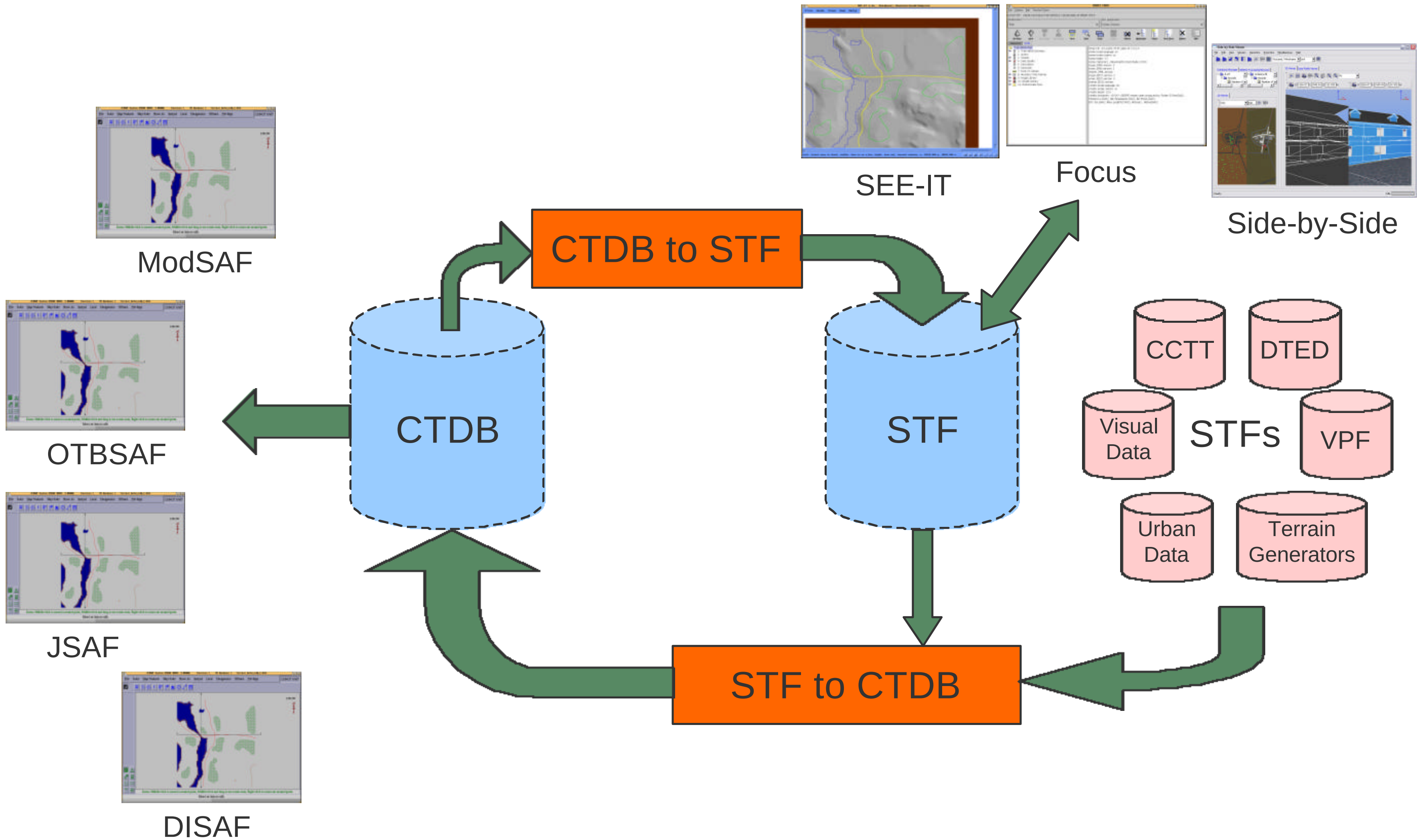
Tool Status: Future Enhancements

- GCS/GTRS support
- MES Building support
- Integrate refinements to TCRS and use future TCRS Tools
  - TCRS language
  - Development of TCRS Checker
- Graphical user interface
  - Easier configuration options
  - Add a help system

Real World Usage

- DTED, VPF native data
  - Create STFs via the DTED to STF and VPF to STF programs.
  - Use STF to CTDB to combine the two STFs into a single CTDB.
  - Converter assumes VPF and DTED are correlated prior to being converted to CTDB.
- VERTS
  - Used TerraTools and STF to CTDB, as well as the DISAF MES insertion process to create urban environment CTDBs with MES structures. (New York and Philadelphia).
- CTDB maintenance and improvement
  - Use the CTDB to STF to extract the CTDB data into an STF.
  - Use SEDRIS based tools, such as SEE-IT, Side-By-Side, and Focus to make improvements to the STF or add new content
  - Use STF to CTDB to create the new and improved CTDB.

Bringing It All Together



Summary

- Introduction to the CTDB format
  - Basic Information
  - Polygon Attribute Table
  - Terrain and Soil Representation
  - Feature Representation
  - MES
- CTDB to STF
  - Software Architecture
  - Data Mappings (CTDB Data Structures to DRM Classes)
  - Geometry/Feature Classification
  - Geometry Coloring
  - GCS
  - Performance
  - Real-World Usage
  - Limitations
  - Software Availability
- STF to CTDB
  - Mapping STF Data to CTDB
  - Overview of Operation
  - Tool Status
  - Real World Usage
- <u>Any Questions?</u>

Where to go from here

- Available mailing lists
  - http://www.sedris.org/
    - Click on E-mail Conference link
      - ctdb@sedris.org
      - tools@sedris.org
- Previous CTDB Papers
  - http://www.sedris.org
    - Click on SEDRIS Papers link
- Previous CTDB Presentations
  - http://www.sedris.org
    - Click on Presentations link
- Request beta versions of software
  - Email: help@sedris.org
- Watch for release of both tools as well as other SEDRIS tools
  - http://tools.sedris.org

References

- J. Smith: “LibCTDB User Manual and Report”
- A. Tosh, J. Campos: “SEDRIS’ing the CTDB Landscape”, Fall SIW 2002.
- K. Wertman, J. Campos: “Using STFs for CTDB Production”, Fall SIW 2002.
- M. A. Pigora, D. Shen, J. Campos: “Innovating with SEDRIS Tools”, Fall SIW 2001
- V. J. Skowronski: “Producing Terrain Databases for Computer Generated Forces using SEDRIS”, Proceedings of the Fall 1998 Simulation Interoperability Workshop, September 1998
- V.J. Skowronski, T. Stanzione: “Using SEDRIS for GCF Terrain Database Generation”