As-Built Modeling Translation Tool

A specific need exists within the "as-built" modeling effort at LLNL to facilitate the transfer of as-built geometries into certain data/image-processing tools, mesh-generation tools, and analysis codes. This project addresses that need.

Project Goals

The objective of this project was to create a simple tool to aid in formatting and converting data between multiple mesh-based geometry formats. The tool links a number of useful finite-element mesh-generation and image-processing tools with analysis codes within LLNL.

Relevance to LLNL Mission

This project directly supports LLNL's as-built modeling effort by enabling the transfer of data between software tools. The basic idea and goal of as-built computational modeling is to incorporate the most representative geometry and material information for a fabricated object into engineering and physics simulations, using nondestructive characterization and metrology techniques to provide the as-built feature information. As-built features might include geometry deviations (such as asymmetries and dents) or material anisotropies and flaws (such as inclusions, voids, cracks, delamination, and ablation regions). The features might originate from the manufacturing process or from the object's being exposed to a certain physical environment during service conditions. By incorporating more representative geometry and material features as initial conditions, the uncertainty in the finite-element simulation results could potentially be reduced, providing a new understanding of the event and object being modeled.

FY2005 Accomplishments and Results

A simple GUI-based Perl translation tool (see figure) was created. The tool eliminates the need to create a special translation script each time a transfer of data between certain software tools is required. This saves the analyst time and effort, and has already proven useful for analysts at LLNL and for outside contractors. Although a limited set of formats and utilities have been currently linked, the utility is easily expandable to include additional formats.

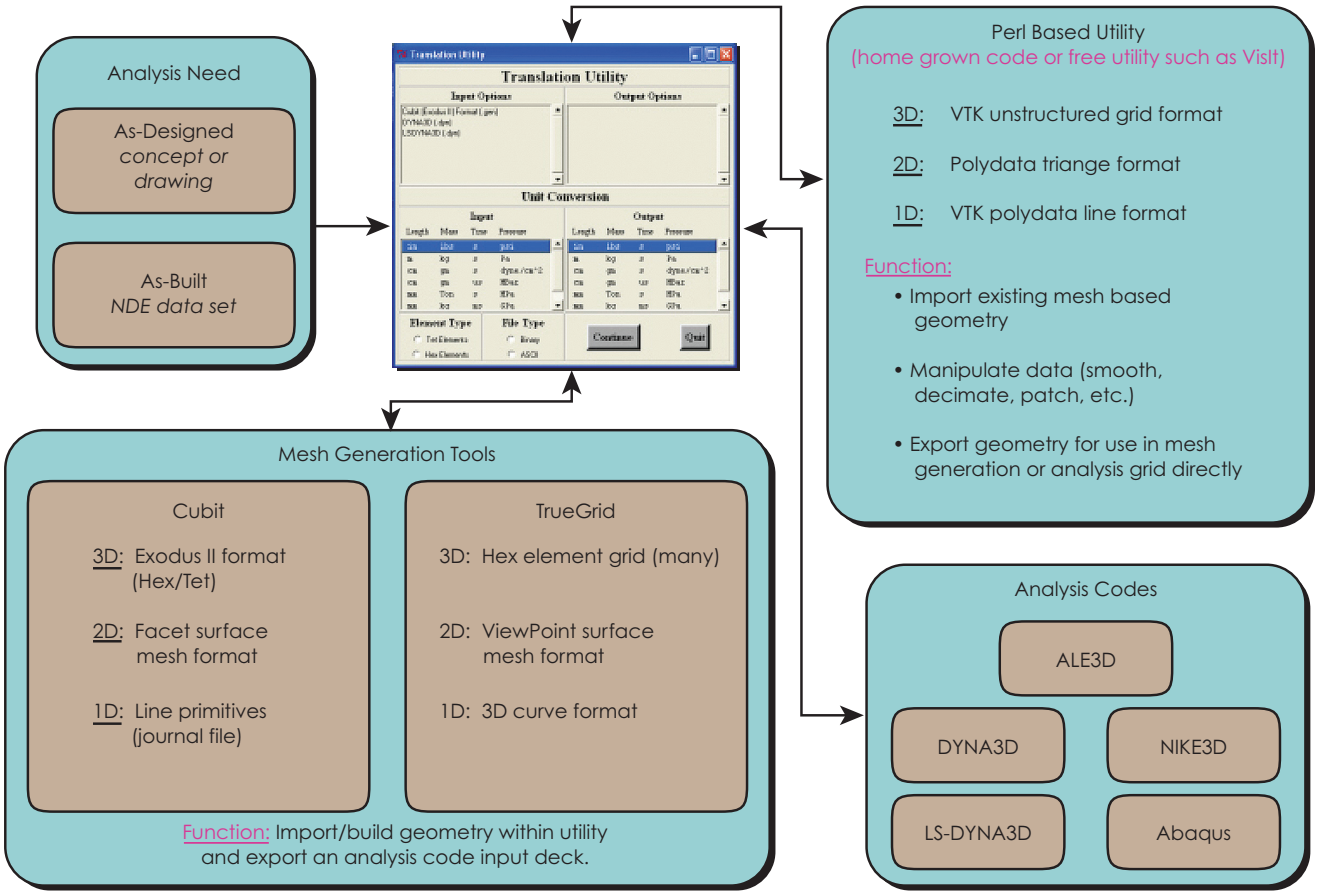
Some of the format conversion paths include:

1. Cubit (Exodus II) format → ALE3D, DYNA3D, NIKE3D, and Visualization Toolkit (VTK);
2. VTK (polydata triangle element) format → Cubit (facet surface mesh) and TrueGrid (ViewPoint surface mesh);
3. VTK (polydata line element) format → Cubit (line journal commands) and TrueGrid (3-D curves);
4. LLNL Dyna3D format → Cubit (Exodus II) format;
5. Abaqus format → ALE3D and VTK formats; and
6. LS-Dyna3D format → ALE3D and VTK formats.

As an example, we have completed the link: VTK mesh- or line-based surface → Cubit or TrueGrid format.

For more information contact
Edwin J. Kokko
(925) 422-4246
kokko1@llnl.gov

This conversion path is useful, since several utilities (Visit, ParaView) exist that can manipulate mesh- and image-based geometry for export to the VTK format. For example, an as-built surface or profile of interest could be isolated within these utilities using commands such as filtering, thresholding, slicing, decimation, and smoothing. The final isolated surface or profile could then be exported from the tool in the VTK format and converted with the translation utility into a useful Cubit or TrueGrid format. Once imported, surface and line data is useful for creating as-built finite-element meshes for subsequent analyses.



Schematic of input and output to the translation utility.