Automated System Notes for the Verification of Interoperability

Duygu

07/09/18

I. Dtest
Tasks:

   A. Reads the ”model-specific” template (header) files (for the parameters, main shape characteristics (compact? convex? etc), system tolerances, algorithm precision and minimum feature sizes, query types supported by C_i)

   B. Serves as an interface that declares the property functions and tolerances and minimum feature sizes as variables

   C. Creates two other scripts: evaluation and configure

   D. Receives interactive information from the user: template file upload and name of the test (verification of interchangeability in our case)

Template File Format

- Authoring CAD system C_i
- System tolerance ϵ_i
- Algorithm precision α_i
- List of queries supported by C_i
- M_i topological class (e.g. manifold, non-manifold, connected, convex etc)
- minimum feature size δ_i

II. Configure
Tasks:

   A. Queries the systems and builds proxy models M^i from M_i

   B. Computes properties on M^is

For now, M^i is an ϵ cover and properties are Hausdorff distance, surface area, volume etc. Ideally, this should be done on a single build environment. I envision this working as a compiler.

III. Evaluation
Tasks:

   A. Compares the properties of M^is and derives results based on ϵ and δ

   B. Creates a report: text file listing the results of the property comparison and some formal statements

Outcome Version 1 for the evaluation script

- System C_i: Rhino 5 (e.g.)
- System tolerances:
  - abs, angular etc..
- Model M_i: Unit cube

- Result: System C_i and C_j are verified as interchangeable with accuracy $c\epsilon$ in applications based on shape similarity or require volume integrations. (This part could be more detailed)