44th AIAA Aerospace Sciences Meeting and Exhibit
9 - 12 January 2006, Reno, Nevada

AIAA 2006-529

The TSTT Mesh Interface

Carl Ollivier-Gooch* Kyle Chand† Tamara Dahlgren† Lori Freitag Diachin†
Brian Fix‡ Jason Kraftcheck§ Xiaolin Li‡ E. Seegyoung Seol¶ Mark S. Shephard¶
Timothy Tautges§ Harold Trease‖

PDE-based numerical simulation applications commonly use basic software infrastructure to manage mesh, geometry, and discretization data. The commonality of this infrastructure implies the software is theoretically amenable to re-use. However, the traditional reliance on library-based implementations of these functionalities hampers experimentation with different software instances that provide similar functionality. This is especially true for meshing and geometry libraries where applications often directly access the underlying data structures, which can be quite different from implementation to implementation. Thus, using different libraries interchangeably or interoperably for this functionality has proven difficult at best and has hampered the wide spread use of advanced meshing and geometry tools developed by the research community. To address these issues, the Terascale Simulation Tools and Technologies center is working to develop standard interfaces to enable the creation of interoperable and interchangeable simulation tools. In this paper, we focus on a language- and data-structure-independent interface supporting query and modification of mesh data conforming to a general abstract data model. We describe the model and interface, and provide programming "best practices" recommendations based on early experience implementing and using the interface.

List of Abbreviations

AI	Adjacency information enum	iter	Iterator over entities	TVT	Tag value type enum
EH	Entity handle	SH	Set handle	Topo	Entity topology enum
ES	Entity set handle	SO	Storage order enum	Type	Entity type enum
ET	Error type enum	TH	Tag handle	VH	Vertex handle

I. Introduction

Creating simulation software for problems described by partial differential equations is a relatively common but very time-consuming task. Much of the effort of developing a new simulation code goes into writing infrastructure for tasks such as interacting with mesh and geometry data, equation discretization, adaptive refinement, design optimization, etc. Because these infrastructure components are common to most or all simulations, re-usable software for these tasks would significantly reduce both the time and expertise required to create a new simulation code.

*Advanced Numerical Simulation Laboratory, University of British Columbia
†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
‡Dept. of Applied Mathematics and Statistics, SUNY Stonybrook
§Parallel Computing Sciences, Sandia National Laboratories
¶Scientific Computation Research Center, Rensselaer Polytechnic Institute
‖Pacific Northwest National Laboratory