ACCESSING CAD GEOMETRY FOR MESH GENERATION

Mark W. Beall[1], Joe Walsh[2], Mark S. Shephard[3]

[1]Simmetrix Inc., Clifton Park, NY., U.S.A. mbeall@simmetrix.com
[1]Simmetrix Inc., Clarkesville, GA., U.S.A. jwalshl@simmetrix.com
[3]Rensselaer Polytechnic Institute, Troy, NY., U.S.A. shephard@scorec.rpi.edu

ABSTRACT

One of the major issues of mesh generation today is access to CAD geometry in an accurate and efficient manner. This paper will provide an overview the process of accessing CAD geometry for mesh generation and will review several of the issues associated with accessing CAD geometry for mesh generation. This paper will also evaluate alternative techniques for accessing CAD geometry and review how these techniques address or do not address the issues related to CAD geometry access for mesh generation. The techniques for CAD geometry access to be reviewed include: Translation & Healing, Discrete Representations, Direct Geometry Access, and Unified Topology Accessing Geometry Directly. The intent of this paper is to provide an overview to the alternative approaches and how they address the specific issues related to accessing CAD geometry for mesh generation. It is not the intent of this paper to provide detailed algorithms related to accessing or repairing CAD data.

Keywords: CAD, geometry, topology, tolerances, design integration, adaptivity, mesh generation, geometry-based, geometry access, Unified Topology Model

1. INTRODUCTION

Automatic and semi-automatic mesh generation has seen dramatic improvements over the last ten (10) years. One of the most important and often overlooked aspects to mesh generation is accessing CAD geometry. The emphasis on analysis in recent years has moved from failure analysis and validation to becoming an active part of the design process. There is a growing demand from manufacturing companies to include performance evaluation based on simulation results earlier in the design process, making simulation an integral part of their design process. To do this in a cost effective manner requires automation of all of the steps involved in performing such simulations from the product design data. Accessing CAD geometry for mesh generation is still one of the major technical issues related to moving simulation forward as an essential ingredient of the design process. This desired ability to move simulation forward in the design process requires a review of current techniques for accessing CAD geometry [1].

This paper will review several of the issues related to CAD geometry access and will evaluate four techniques for CAD geometry access as follows: 1) Translation & Healing, 2) Discrete Representations, 3) Direct Geometry Access, 4) Unified Topology Accessing CAD Geometry Directly.

2. CAD GEOMETRY

CAD systems and their geometric representations have been around for quite some time. Almost all CAD systems have evolved into similar representations for their models. This representation often includes feature based data and a resulting B-Rep instance or a B-Rep model. The B-Rep model consists of much more than just geometry, and indeed one of the major problems in accessing CAD geometry has been due to an oversimplification of what constitutes a valid B-Rep Model.

B-Rep models contain geometry (shape), topology (how things are connected), and tolerances (how closely do they actually fit together). This combination of model data is then accessed by the CAD systems methods to define a valid B-Rep model. Therefore, a valid B-Rep model should be considered to consist of geometry, topology, tolerances and methods used by the CAD system it was defined within [2].

CAD systems often use relatively large tolerances on an entity-by-entity basis to provide robustness to model operations. This approach is referred to as variable tolerances or tolerant modeling by different CAD systems. The use of these large variable tolerances produces gaps and overlaps in the geometry and topology of the CAD system B-Rep model as illustrated in the simple (and extreme) example in Figure 1.

The algorithms used in the CAD system modeling engines are written to deal with these tolerances in a consistent manner and they do not see the gaps or overlaps.

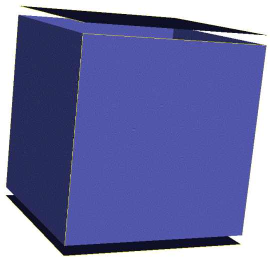


Figure 1. Large / variable tolerances result in gaps and overlaps

Geometric modeling kernels such as ACIS, Granite and Parasolid are often used to supply the methods and model representations used by CAD system modeling engines. CAD systems that use a common geometric modeling kernel also share common methods for evaluating tolerances and the validity of a B-Rep model. These methods can be accounted for directly in the mesh generation process in a consistent manner using information easily provided by the CAD system API [2], [3].

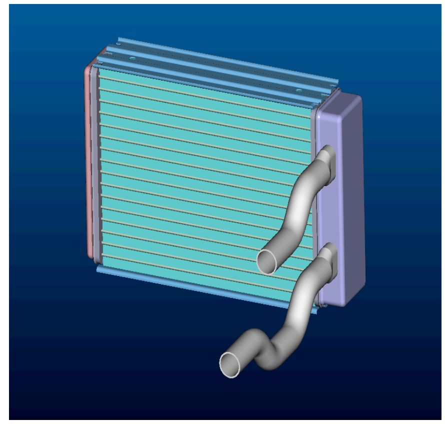
3. GEOMETRY RELATED ISSUES FOR MESH GENERATION

There are several issues associated with effective and efficient access of CAD geometry for mesh generation. This section will provide a quick overview of several of the major issues and the ramifications that this issues have on mesh generation. A detailed review of these issues is beyond the scope of this paper. Specifically excluded from this paper are model abstraction or idealization for analysis and domain decomposition.

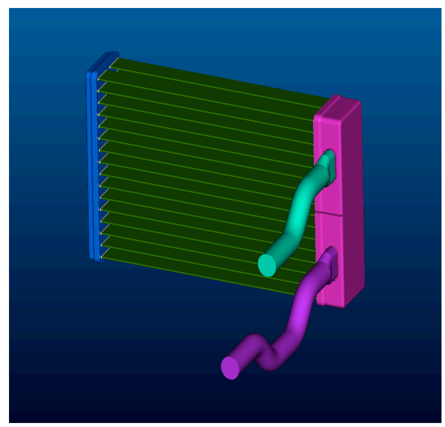
3.1 Understanding the Analysis Requirements

The first major issue with CAD geometry access for mesh generation is the need to understand the analysis requirements. The appropriate mesh and geometry to be used for meshing is a function of the analysis to be performed and the desired accuracy [4]. There does not exist an optimal mesh independent of the analysis to be performed. *A-prioi* element shape quality test have often been used as a misleading indicator of a good mesh independent of the analysis to be performed or the accuracy desired. The appropriate mesh is one that produces the desired accuracy for the problem to be solved. In practice this is only achievable through adaptivity.

Different types of analyses require different instances of the geometry to capture the physics. For example, we can perform a dynamic structural response analysis and a Computational Fluid Dynamics (CFD) analysis on the same part. The dynamic structural response analysis requires the solid geometry of the part while the CFD analysis requires the geometry of the cavities through which the fluid will flow. This simple illustration of different use of geometry representations is illustrated in figure 2.



Dynamic structural response analysis requires solid geometry of the part.



While CFD analysis requires geometry of the flow cavities.

Figure 2. Different analysis require different geometric representations

Physics simulations such as external flow, electromagnetics, and radiation are actually concerned with the volume not occupied by the part.

Different types of analysis also require different resolutions of mesh to achieve the desired accuracy on a particular design.

3.2 Defeaturing

Defeaturing is one of the most complex issues associated with CAD geometry access for mesh generation. Indeed one of the major issues that the CAD and CAE software industries have encountered is developing a consistent definition of a feature. For the purposes of this paper we will classify features into two main groups.

The first group of features will be called "intended features". *Intended features* are features that were explicitly defined as features in the model that drive the resulting geometry. In this case a feature-based modeling system was used to create a model which contains *intended features*. *Intended features* can only be created by feature-based modeling systems and can be suppressed by the original modeling system.

The second group of features will be called "artifact features". *Artifact features* are features that are created indirectly by the modeling process. One example of *artifact features* is the creation of engineering features such as holes by a modeling system that is not feature-based. The second example of *artifact features* is the creation of recognizable patterns of geometry / topology data that create a valid design model but also create difficulties associated with mesh generation. Artifact features can be created from any modeling system and cannot be suppressed in the original modeling system.

Part of the complexity associated with CAD geometry access for mesh generation is due to the fact that historically analyses are performed too late in the design process and the design model contains more details than are appropriate for analysis. Moving the analysis earlier in the design process will help to reduce, but will not remove, the need for defeaturing. Since multiple analysis types may be required for any design state there remains a need for defeaturing to various levels to support the range of analysis to be performed.

One of the most common unwanted *artifact features* encountered in CAD data are "slivers". *Slivers* can be described as very small *artifact features* that are larger than the geometric tolerances of the CAD System modeling engine but extremely small with respect to the model size. These very small *artifact features* can provide problems to mesh generation algorithms and are meaningless to the analysis [5]. *Slivers* are introduced into models to maintain validity and integrity of the model. Native models contain far fewer *slivers* than translated models. A very common method of healing or repair algorithms used in translation is to introduce *slivers* to resolve gaps, overlaps and tangency conditions.

Modeling engines and healing algorithms may also introduce an large number of faces into the model to ensure that the model is valid. This often occurs in blend and chamfer regions and in areas of similar surface curvature or near tangent conditions. These additional faces may over-constrain mesh generation and one approach is to combine faces into a single larger logical face. This approach has had significant success but typically relies on user input to specify which faces to combine. Extra faces are another type of *artifact feature*.

Another common unwanted *artifact feature* type is "small" model features. Small model features can be described as *artifact features* that are very large with respect to the geometric tolerances but small with respect to the local target mesh size. This definition of *small* features indicates that the classification of a *small* feature is a function of the target element size and accuracy desired. The actual definition of the *small* sizing with respect to target mesh size can vary with each analysis to be performed. Some typical values for *small* features are less than 25-30% of the target mesh size. This definition also allows for support of an adaptive representation of geometry used for meshing as part of the mesh adaptivity process that we will discuss further later in this section.

Slivers may be re-classified as a special case of *small* features that will remain *small* through all possible target mesh sizes. *Small* features are another type of *artifact feature*. The issue of dealing with *small* geometric features in the mesh generation process has been discussed in various references [6], [7]. An example of a *small* feature and its potential impact on mesh generation is illustrated in Figure 3.

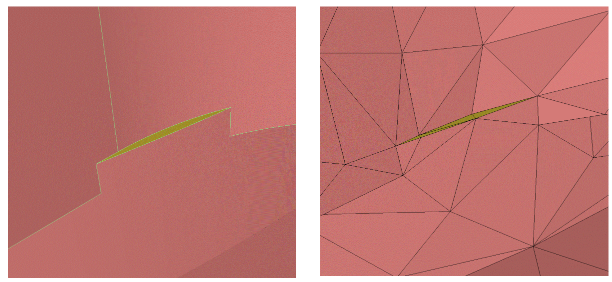


Figure 3. *Small* feature

CAD models may include geometric features that are important for design but are irrelevant for the simulation to be performed. These unwanted features can be classified as "simple" features and "complex" features. These features can be suppressed by the CAD system if and only if they were *intended features*.

Simple features can be described as features which when suppressed or removed refer back to a single parent face on the B-Rep model. *Simple* features may be *intended features* or *artifact features* but are most likely *intended features* from a feature-based modeling system. *Simple* features are defined in terms of the topology of their base features rather than their size. Examples of simple features are illustrated in Figure 4.

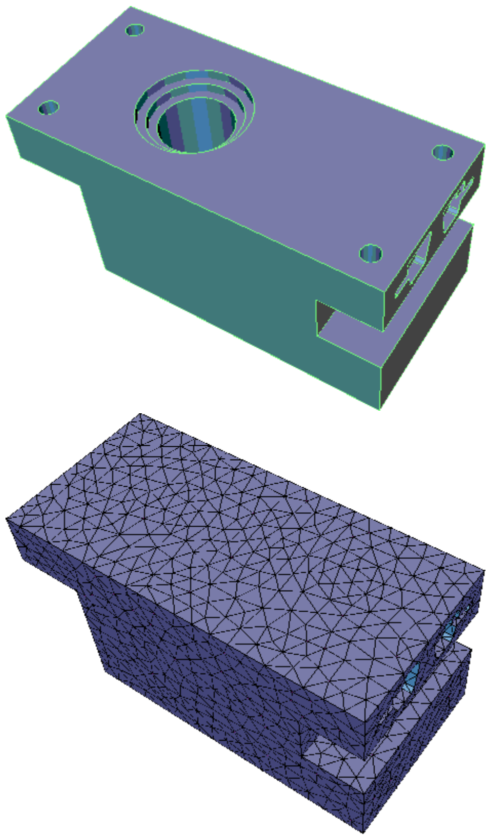


Figure 4. ***Simple*** **features on top face**

Complex features can be described as features that are not *simple*. *Complexe* features may be *intended features* or *artifact features* but are most likely *intended features* from a feature-based modeling system These features include a variety of features as follows:

- Features whose base feature spans across multiple faces.
- Features whose base features need to be extended for feature removal or suppression such as fillets and chamfers.
- Features that interfere with other features.

Complex features are the largest challenge to deal with in defeaturing. If these features are not *small* with respect to target mesh size, careful consideration should be given regarding why these are being defeatured and the impact on accuracy. If these features are *small* then they can be treated as *small* features independent of their complexity. For *complex* features that need to be removed or suppressed that are not small a thorough understanding of the feature data is required and it usually best to suppress these in the CAD system prior to geometry access.

3.3 Tolerances and Methods for Evaluating Tolerances

Understanding tolerances and methods for evaluating tolerances plays an important role in accessing CAD geometry for mesh generation. One of the key areas influenced by tolerances and their associated methods is that of tangencies and near tangencies. The methods used in CAD system modeling engines are written to deal with tolerances in a consistent manner. These methods are not available outside of the CAD system modeling engines, therefore, translated data introduces "dirty" geometry.

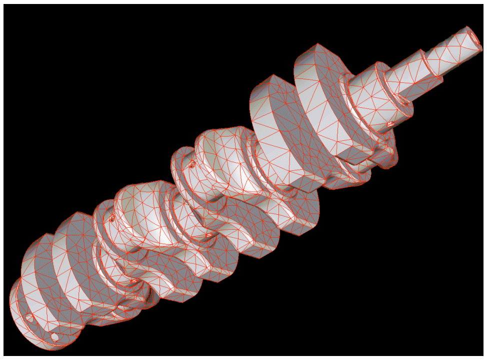
3.4 "Dirty" Geometry

Dirty geometry has been one of the most nagging issues related to geometry access. *Dirty* geometry consists of gaps, overlaps and other incompatibilities in the model preventing the model from being valid. These incompatibilities do not exist in the native CAD system and are introduced from translating the native CAD geometry to another format. Differences in representations, methods and tolerances between modeling engines create *dirty* geometry. Translators must then heal or repair the geometry to represent it as a valid model in the non-native system [5], [8], [9]. Note that without knowledge of the modeling system tolerances and methods, there is no *a-priori* means to ensure a healing process will successfully recover the correct model representation.

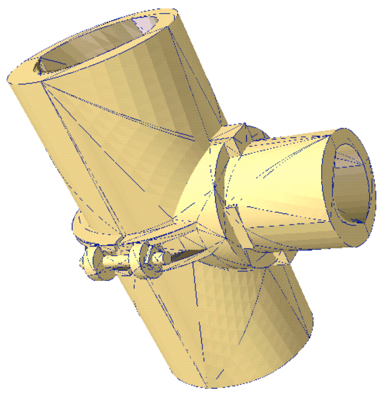
3.5 Support for Curved Meshing

The previous issues associated with geometry access have focused on ensuring the correct geometry representation and level of detail in the geometry be used for mesh generation. The next three issues deal with ensuring that the geometry access can support key mesh generation functionality. The first mesh generation functionality to be considered is curved meshing. Curved meshing involves the ability to create curved mesh edges and faces that have the level of geometric approximation needed to ensure that as the simulation results are improved by the introduction of higher-order equation approximations (e.g., high-order finite elements), the geometric approximation errors do not control the solution accuracy. The ability to properly curve the mesh entities occurs arises as soon as higher than linear basis functions are used and, as demonstrated by simple example in [10], the order of geometric approximation needed to be increased as the basis order increases.

In the simplest cases, the appropriate curved meshes can be created by moving mesh on the boundary of the model to the "closest" location on the model geometry. However, even in the simplest, and common, case of quadratic h-type finite elements (see example at the top of Figure 5), a more complex algorithm is required to ensure the elements can be properly curved [11]. The complexity of the curved mesh generation process increases further in the case of p-version methods where coarse meshes, such as the example at the bottom of Figure 5, must have higher order geometric approximations.



Initial coarse "h" type curved mesh



Very Coarse "p" type curved mesh

Figure 5. Curved Meshing

3.6 Support for Curvature Based Mesh Refinement

The next meshing functionality to be considered as desirable to be supported is curvature based mesh refinement. This meshing functionality provides automatic refinement of the mesh based on the underlying geometry curvature. The benefits of this functionality are: 1) the ability to capture the geometry with a considerable smaller number of elements and/or grid points and 2) resulting improvement in mesh quality in areas of rapid geometric changes. Figure 6 illustrates the benefits of curvature based mesh refinement.

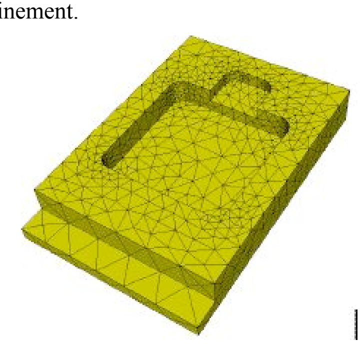
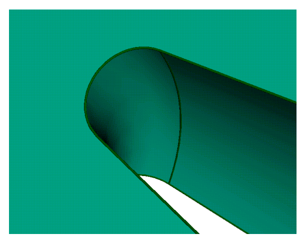


Figure 6. Curvature Based Mesh Refinement

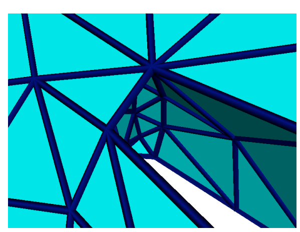
3.7 Support for Geometry Based Mesh Adaptivity

The final mesh generation functionality to be considered, in this paper, as an issue for geometry access is the support for geometry based mesh adaptivity. This functionality involves the ability of the adapted mesh to adhere to the original geometry as illustrated in figure 7 and requires access to the original geometry to be present. Mesh adaptivity that does not adhere to the geometry is limited by the initial mesh geometric approximations and can provide results that are meaningless. For example, Figure 7 is a close-up of a geometric feature in an accelerator cavity geometry where the simulation procedures must provide highly accurate estimates of the electrical and magnetic losses. The sensitivity of the results to the local geometric shape is so high that if the mesh geometric approximation did not improve as the adaptive simulation process continued, the results obtained would have been not just a poor approximation, but meaningless.

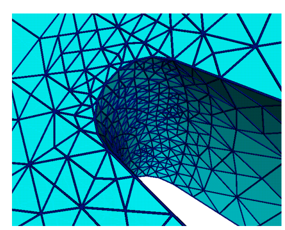
In many problems of interest the mesh edges and faces are of the same size as the small geometric features that are often critical to the analysis, such as the accelerator cavity. In these cases, the simple movement of new nodes introduced during refinement to the curved model surfaces can yield invalid elements. The algorithms needed to effectively deal with these situations include must include general mesh modification operations and a control algorithm that ensures the procedure is progressing in a positive manner [12].



Solid model detail of complex blend feature



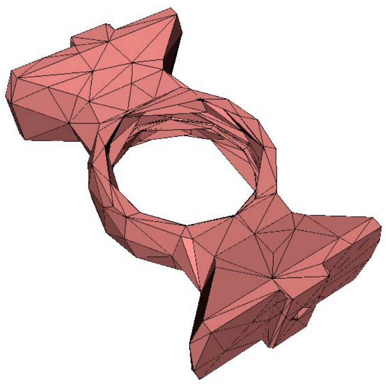
Initial coarse mesh



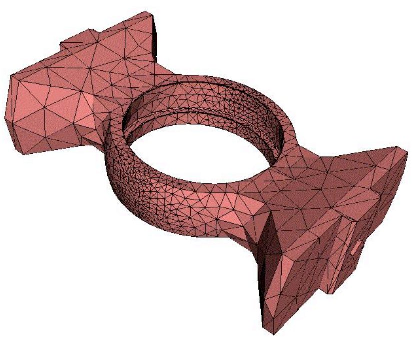
Adaptive mesh adheres to initial geometry

Figure 7. Geometry based mesh adaptivity

The advantages of geometry based mesh adaptivity include: 1) the ability to start with coarser initial meshes and, 2) the ability to ensure that the resulting model adheres at an appropriate level of accuracy to the design geometry. An additional benefit that may not be apparent is the combination of geometry based mesh refinement with the *small* feature defeaturing as a function of target mesh size. This can result in adaptive geometry representation for mesh adaptivity where *small* features are ignored in the initial mesh and accounted for as a function of target mesh size in each stage of the mesh adaptivity process. This combined approach dramatically reduces the defeaturing requirements associated with geometry access for mesh generation and allows for initial coarse meshes of detailed geometric models. Figure 8 illustrates an example of this combined approach to adaptive geometry representation.



Initial coarse meshes approximates *small* features



Adaptive mesh accurately accounts for *small* features

Figure 8. Adaptive geometry representation

3.8 Evolving Geometry Problems

There are a number of situations where the model shape and topology can evolve during the simulation. When the simulation is performed using Lagrangian type analysis and there are large deformations and/or model fracturing, it is often necessary to update the domain and mesh several times during the simulation (e.g., in fragmentation simulations [13] or metal forming 14]). In these situations the model topology and shape must be updated based on the simulation results. Even in the case where the original geometric model was defined in a CAD system, it is most likely not desirable to continue to use the original CAD system to update the CAD model. This is because the new geometric information available from the simulation is limited to node point coordinates on the mesh facets and most CAD systems do not effectively support such geometry updates.

An important aspect of properly updating the geometric model for these cases is to update the model topology based on the simulation information and to associate the appropriate collections of mesh edges and mesh faces with the resulting model edges and faces to use in the subsequent definition of shape information. Algorithms to do this based on mesh based geometry parameters and/or simulation contact or fracture information have been developed [15], [16], [14]. Once the model topology has been defined, the geometric shape information can be defined directly in terms of the mesh facets, or can be made higher order using subdivision surfaces [17], [16] or higher order triangular patches [18] [19]. Reference [14] provides a description of an automated adaptive medal forming procedure where the updated geometric model is defined based on the simulation information and higher order updated shapes of the edges and surfaces are defined by subdivision patches applied on a model entity level.

3.9 Integration of Simulation in the Design Process

Integration of simulation in the design process is a driving factor for improved geometry access for mesh generation and support of this integration should be considered as a major issue when considering geometry access. This integration allows for simulation to be an integral part of the design process and requires use of the native CAD system geometry as the geometry source to allow for effective reuse through multiple design iterations. Mesh generation needs to access the current design state and evolve with the design [20]. Automatic meshing and geometry based mesh refinement are fundamental requirements to ensure efficiency and accuracy. Integration of simulation in the design process also requires sophisticated management of simulation attributes to support design change insensitivity for simulation.

3.10 Multiple CAD Geometry Sources

The geometry access issues discussed so far in this paper are limited to a single CAD system. These issues are further complicated by the need to support multiple CAD systems. Each CAD system modeling engines uses different representations for geometry and topology and different tolerances and methods for evaluating tolerances. Direct interface utilities to multiple CAD systems is both complex and expensive to develop and support. Modeling kernels such as ACIS, Granite and Parasolid help to reduce the scope of this problem.

Commercial software vendors need to provide support for multiple CAD systems to properly support their customer base. It should also be noted that large-scale design environments and processes typically consist of multiple CAD systems both internally within a company and throughout the supply chain.

4. TECHNIQUES FOR ACCESSING CAD GEOMETRY FOR MESH GENERATION

There are several techniques currently used and being developed to address the geometry access issues outlined in this paper. The techniques used for geometry access can be classified into four major approaches as follows:

- Translation & Healing
- Discrete Representations
- Direct Geometry Access
- Unified Topology Accessing Geometry Directly

4.1 Translation and Healing

Translation and Healing has historically been the most commonly used technique for geometry access. The translation may involve use of standard file formats or direct translators.

IGES does not address issues with representations, global tolerances, features, tolerancing or tolerance methods and typically results in *dirty* geometry [5]. Standards such as VDAFS and STEP do address issues with representations and global tolerances but do not address features, tolerancing or tolerance methods and often results in *dirty* geometry (typically cleaner than IGES).

Many companies have invested millions to resolve the translation related issues (ITI, Elysium, Spatial, TransMagic, CAD-CAMe, TTI, TTF, …) An entire interoperability industry has evolved to attempt to address the issues of Translation and Healing. Progress has been made but the Translation and Healing process is still not reliable or robust. The fundamental issue of differing native tolerance methods has not been addressed.

Evaluation of Translation and Healing as related to geometry access issues presented in this paper is as follows:

- Defeaturing is difficult since *intended feature* information is lost in translation and unwanted *artifact features* may be created.
  - Feature-based translators attempt to reproduce models from feature representations but do not address tolerance methods and may fail to rebuild models or introduce *slivers* and *small* features.
  - Healing typically introduces *slivers* and *small* features to resolve *dirty* geometry.
  - Non feature-based translators require explicit feature removal.
  - Feature suppression with non feature-based translators requires feature recognition algorithms.
- Translation & Healing introduces *dirty* geometry due to differences in CAD systems modeling engines representations, tolerances and methods.
- The resulting geometry representation typically can support curved meshing, curvature based refinement and geometry based mesh adaptivity on modified representation.
  - It is possible to support adaptive geometry representation on modified representation with *small* feature recognition.
- The ability to support evolving geometry is limited by the geometry representation available.
- The integration of simulation in the design process is not effectively addressed.
- Differences in algorithms and tolerances between modeling engines make it impossible to exactly exchange data between them. Results and robustness vary dramatically with different CAD systems.

4.2 Discrete Representations

The Discrete Representations technique is based on the generation of a faceted model by the CAD system and accessing the resulting faceted model for mesh generation. This is most commonly done based on simple facets generated by the CAD system faceter but may also use subdivision surfaces [17], [16] or higher order triangular patches [18].

This technique is often used to attempt to eliminate *dirty* geometry and to resolve differences between different CAD systems. There are some remaining concerns regarding robustness since the simple facet representations are designed for visualization and may not close as illustrated in figure 9. These facet representations are often done on a face-by-face basis and may not be incompatible across face boundaries.

The successful use of the simple facets in Discrete Representations technique is highly dependent on the faceter used by the originating CAD system. All Discrete Representation techniques result in an approximation of the geometry and do not retain the *intended feature* data, and geometry of the CAD model.

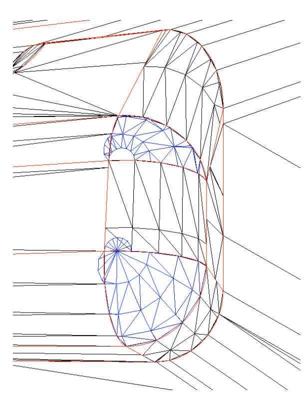


Figure 9. Facet representations from major CAD system modeling engines may not close

Evaluation of Discrete Representations as related to geometry access issues presented in this paper is as follows:

- Defeaturing of any type is difficult since all *intended feature* information is lost.
- Simple facet representations are designed for visualization and may still have some problems with *dirty* geometry.
- Simple facet representations cannot support curved meshing, curvature based refinement and geometry based mesh adaptivity.
    - More sophisticated discrete representations such as subdivision surfaces and higher order triangular patches can support an approximate version curved meshing, curvature based refinement and geometry based mesh adaptivity.
    - It is difficult to support adaptive geometry representation on modified representation with *small* feature recognition.
- The definition of evolving geometry can be supported.
- The integration of simulation in the design process is not effectively addressed.
- Handles data from different systems in a consistent manner but results may vary dramatically due to differences in CAD System faceters.

4.3 Direct Geometry Access

Direct Geometry Access is a technique that is growing in popularity based on accessing CAD geometry directly through CAD system toolkits such as CATIA CAA and Pro/Toolkit [21]. Use of the CAD system toolkits requires that a seat of the CAD system is available for geometry access.

Since many CAD systems use geometric modeling kernels this approach can also achieved by licensing the same geometric modeling kernel as the CAD system and accessing the geometry through the modeling kernel APIs [1], [3], [22], [23].

The main theme of this approach is to leave the data in the native modeling engine and to use that native modeling engine to access geometry so that the native tolerances and methods are used for geometry access and wherever possible the *intended feature* data is retained.

Evaluation of Direct Geometry Access as related to geometry access issues presented in this paper is as follows:

- Defeaturing is an issue for *artifact features* that cannot be suppressed.
    - *Small* features, *slivers* and multiple faces cannot be suppressed.
- Native geometry is not *dirty*.
- Can support curved meshing, curvature based refinement and geometry based mesh adaptivity.
    - Adaptive geometry representation with *small* feature recognition is extremely difficult (if not impossible) to support.
- The ability to support evolving geometry is limited by the geometry representation available.
- The integration of simulation in the design process can be effective addressed with unique solutions for each CAD modeling source.
- Requires multiple direct interfaces for a broad range of geometry support.
    - Each CAD system has a different geometry and topology representation to interrogate for meshing.
    - Each CAD system has different tolerances and methods to understand.
    - Each CAD system has a different toolkit for accessing geometry and topology data.

4.4 Unified Topology Accessing Geometry Directly

The final geometry access technique to be considered is Unified Topology Accessing Geometry. This is a natural extension of the Direct Geometry Access technique with enhancements to overcome the shortfalls of that technique (especially associated with multiple CAD sources and defeaturing of *artifact features*. This approach is based on an abstraction of the geometry that allows multiple sources of geometry to be treated the same by the mesh generator [1], [3], [22], [23]. For the purposes of this paper this abstraction of the geometry will be referred to as the Unified Topology Model.

The Unified Topology Model is a representation of the model for simulation purposes that retains it connection to the original CAD system geometry and topology. This approach provides a separate topology data structure that allows for multiple forms of defeaturing while retaining the original geometry & topology. This approach also facilitates the use of geometry from multiple sources.

The geometry is directly accessed from the native modeling system as per the Direct Geometry Access technique, however, a common description of the topology is created that is well suited for mesh generation. This Unified Topology Model accounts for the topology of the original modeling systems and enhances this representation to make it more suitable for analysis. These enhancements may include; support for multi-dimensional models, non-manifold model (extremely useful for assemblies), defeaturing of unwanted features, and support for models from multiple CAD sources for a single analysis.

One important aspect of the Unified Topology Model is to maintain a relationship between the Unified Topology Model and the topology of the original CAD model. This may be a one to one relationship, or a one to many relationship. Maintaining these relationships allows the Unified Topology Model to be modified for analysis without affecting the underlying CAD model while still maintaining the Direct Geometry Access for all geometric queries.

One example of a Unified Topology Model is the Simulation Modeling Suite provided by Simmetrix, another example is the CGM provided by Sandia National Laboratories. In the Simmetrix example the Unified Topology Model builds on top of the CAD topology to present a standard representation for all modeling sources (non-manifold topology similar to Radial Edge Data Structure [24]). The Unified Topology Model is built from the CAD topology and geometric queries are passed through to the CAD system via direct access to APIs or modeling kernels. The implementations by Simmetrix and Sandia also support discrete geometry as a modeling source. The resulting Unified Topology Model used is illustrated in Figure 10.

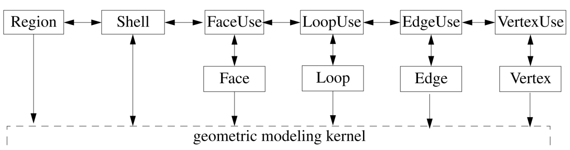


Figure 10. Unified Topology Model

Evaluation of Unified Topology Accessing Geometry as related to geometry access issues presented in this paper is as follows:

- Allows for various forms of defeaturing.
  - *Slivers* and *small* features can be addressed as a function of global and local target mesh sizes. Figure 11 illustrates the effect on meshing results related to defeaturing of the *small* features illustrated in Figure 3.

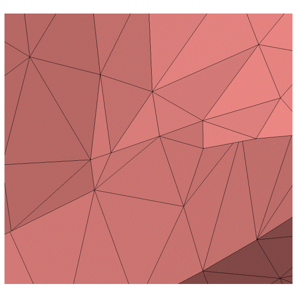


Figure 11. Small feature removed from Unified Topology Model

  - *Simple* features can be suppressed in Unified Topology Model for meshing purposes.
  - *Complex* features may be addressed either by suppression of *intended features* in the CAD system or as *small* features in the Unified Topology Model.
- Uses native system tolerances and methods.
  - Native geometry is not *dirty*.
- Curved meshing, curvature based refinement and geometry based mesh adaptivity can be supported.
  - Can support adaptive geometry representation with *small* feature recognition.
- Creation of new topology in the Unified Topology Model based on a discrete geometry basis provides support for evolving geometry problems.
- Proven effective to address issues related to integration of simulation in the design process.
  - Used in large Simulation-Based Design initiatives and commercial CAE Software (Visteon, John Deere, Blue Ridge Numerics, CFD Research Corporation, ESRD, Coventor, PVM Corporation and many others)
- Provides a single interface for a broad range of geometry support.
  - Geometry abstraction layer handles all CAD systems specific issues.
  - Mesh generation algorithms access a consistent Unified Topology Model.

5. SUMMARY

The desire to use simulation as an integral part of the design process has necessitated an evaluation of the issues and techniques associated with CAD geometry access for mesh generation. A broad range of issues was highlighted in this paper and four techniques for CAD geometry access were reviewed with respect to these issues.

Translation and Healing was the initial technique reviewed and was found to lack the reliability and robustness necessary to support design/analysis integration. The Translation and Healing technique does not address several of the geometry access issues outlined.

The second technique reviewed was Discrete Geometry Representations. This technique does address some of the geometry access robustness issues but does not address well those issues related to feature representations, curvature based meshing and design integration.

The third technique reviewed was Direct Geometry Access. This technique does address many of the geometry access issues but does not address well those issues related to defeaturing of *artifact features* and multiple CAD systems.

The final technique reviewed was Unified Topology Accessing Geometry Directly. This technique provides an effective means to address to the geometry access issues outlined in this paper. The Unified Topology Model Accessing Geometry Directly technique is the most flexible technique for addressing issues related to accessing CAD geometry for mesh generation.

Unified Topology Accessing Geometry Directly can provide a single environment to effectively deal with integration to CAD from multiple sources, along with integration with various discrete models, and defeaturing of *artifact features* providing a firm foundation for design/analysis integration.

REFERENCES

[1] Shephard, M.S., Georges, M.K. and de Cougny, H.L., "Geometric model interactions required to support a fully automatic mesh generator", SCOREC report 25-1993, RPI, Troy, NY, 1993.

[2] Braid, I., "A History of Geometric Modeling", *Spatial Tech-Ex*, pp. 1-1 – 1-17, 1991.

[3] Shephard, M.S., and Georges, M.K., "Reliability of Automatic 3-D Mesh Generation", *Comp. Meth. Appl. Mech. and Engng*., 101:443-462, 1992.

[4] Walsh J.L. "Exposing the Myths of Design to Analysis Data Exchange", *Proc. ABAQUS User's Conference*, pp 659-672, 1993

[5] Butlin, G., Stops C., "CAD Data Repair", *Proc. 5th Int. Meshing Roundtable,* pp. 7-12, 1996.

[6] Shephard, M.S., Beall, M.W. and O'Bara, R.M., "Revisiting the elimination of the adverse effects of small model features in automatically generated meshes", *Proc. 7th International Meshing Roundtable '98*, SAND 98-2250, Sandia Nat. Labs., Albuquerque, NM, pp. 119-131, 1998.

[7] Dey, S., Shephard, M.S. and Georges, M.K., "Elimination of the adverse effects of small model features by local modifications of automatically generated meshes", *Eng. With Computers*, 13(3):134-152, 1997.

[8] Mezentsev, A.A. and Woehler, T., "Methods and algorithms of automated CAD repair for incremental surface meshing", *Proc. 8th Int. Meshing Roundtable,* Sandia report SAND 99-2288, pp. 299-309, 1999.

[9] Ribo, R., Bugeda, G. and Onate, E., "Some algorithms to correct a geometry in order to create a finite element mesh", *Computers and Structures*, 80:1399-1408, 2002.

[10] Luo, X., Shephard, M.S., Remacle, J.-F., O'Bara, R.M., Beall, M.W., Szabó, B.A. and Actis, R., "p-Version Mesh Generation Issues", 11th International Meshing Roundtable, Sandia National Laboratories, pp. 343-354, 2002.

[11] Dey, S., O'Bara, R.M. and Shephard, M.S., "Curvilinear mesh generation in 3D", *Computer-Aided Design*, 33:199-209, 2001

[12] Li, X., Shephard, M.S. and Beall, M.W., "Accounting for curved domains in mesh adaptation", I*nternational Journal for Numerical Methods in Engineering*, 2002.

[13] Pandofi, A. and Ortiz, M., "An efficient procedure for fragmentation simulations", *Engng. With Computers,* 18(2):148-159, 2002.

[14] Wan, J., Kocak, S, Shephard, M.S. and Mika, D., Automated adaptive forming simulations", submitted to *12th Int. Meshing Roundtable*, 2003.

[15] Krysl, P. and Ortiz, M., "Extraction of boundary representation from surface triangulations", *Int J. Num. Meth. Engng,* 50:1737-1758, 2001.

[16] Lee, C.K., "Automatic metric 3-D surface mesh generation using subdivision surface geometry model. Part 1: Construction of underlying geometric model", *Int J. Num. Meth. Engng,* 56:1593-1614, 2003.

[17] Cirak, F., Ortiz, M. and Schroder, "Subdivision surfaces: a new paradigm for thin shell finite-element analysis", *Int J. Num. Meth. Engng,* 47:2039-2072, 2000.

[18] Owen, S.J. and White D.R., "Mesh-based geometry: A systematic approach to constructing geometry from a finite element mesh", *Proc. 10th Int. Meshing Roundtable,* Sandia report SAND 2001-2967C, pp. 83-96, 2001.

[19] Owen, S.J., White D.R. and Tautges, T.J., "Facet-based surfaces for 3-D mesh generation", *Proc. 11th Int. Meshing Roundtable*, pp. 297-311, 2002.

[20] Walsh J.L., "Geometrically Associative Analysis Modeling", *Spatial Tech-Ex*, pp 9-1 – 9-16, 1991.

[21] Merazzi, S., Gerteisen, E.A. and Mezentsev, A., "A generic CAD-mesh Interface", *Proc. 9th Int. Meshing Roundtable,* Sandia report SAND 2000-2207, pp. 361-369, 2000.

[22] Tautges, T.J., "The common geometry module (CGM): A generic, extensible geometry interface", *Proc. 9th Int. Meshing Roundtable,* Sandia report SAND 2000-2207, pp. 337-359, 2000.

[23] Shephard, M.S., "Meshing environment for geometry-based analysis", *International Journal for Numerical Methods in Engineering*, 47(1-3):169-190, 2000.

[24] Weiler, K.J., "The radial-edge structure: A topological representation for non-manifold geometric boundary representations", *Geometric modeling for CAD applications, North Holland*, pp. 3-36, 1988.