The Integration of CAD Systems and Unstructured Mesh Generators

S.P. Kopyssov, V.N. Rychkov, A. B. Ponomarev

Institute of Applied Mechanics UB of RAS,

222 Gorky St. Izhevsk, Russia 426001

{kopyssov,an}@udman.ru

1 Introduction

Finite element analysis begins with the problem statement and the geometry design. To make the computational grid it is necessary to describe the geometry in one way or another, to generate the elements and to set the boundary condition. The last distinguishes the computation model from the geometric one. It is sufficiently difficult problem to form the input data for mesh generator. As rule, decoding of graphic information files (*.dwg, *.sat, *.igs etc) is used. In addition to graphic format support CAD systems have the interfaces that enable to operate with the graphic objects programmatically (programmer interfaces). Thus, it is rational to reuse graphic 3D design to make the computational grid with help of any mesh generator met user's requirements.

The integration of CAD systems and object-oriented mesh generators is examined in present work. The mismatch of their geometric kernels is the main problem. Geometric kernel is the library of common mathematical functions that describe and store 2D and 3D objects. Unlike CAD system, mesh generator requires exhaustive geometric data included both solid and surface definitions. CAD systems and mesh generators are brief reviewed below and then the integration of Mechanical Desktop and GRUMMP mesh generator is discussed.

2 Review of 3D Modeling Systems

There are many CAD systems (Computer Aided Design) [1]. Numerical mathematics needs those of them that are intended for complex 3D design. Geometric kernel and open interface are the main features of such CAD systems. They allow CAD systems to be used in calculations.

2.1 Geometric Kernels

A geometric kernel is the main part of a CAD-system. The geometric data the kernel of CAD-system operates is mainly destined to visualize complex design. Now there are two widely used geometric kernels: Parasolid and ACIS.

Parasolid is the most hide-speed object-oriented geometric kernel. Parasolid provides solid and integrated surface design, tesselated and laminated modeling. CAD/CAM-systems based on Parasolid: SolidWorks (SolidWorks Corp.), SolidEdge and Unigraphics Modeling (Unigraphics Solutions Inc.), IronCAD (VDS) etc.

ACIS is the object-oriented geometric C++ library, including wire-frame, surface and solid models, wide range of geometric operations to design and operate complex model. Besides, ACIS allows surface and solid models to join to each other. Therefore ACIS takes priority of Parasolid intended to solid modeling. CAD/CAM-systems based on ACIS: AutoCAD 2000, Mechanical Desktop and Autodesk Inventor (Autodesk Inc.), IronCAD (VDS), T-FLEX CAD (Top Systems) etc.

2.2 CAD Programmer Interfaces

CAD systems that widely used in the integrated systems have open programmer interface. CAD programmer interfaces are evolving constantly. Now three stages can be picked up.

1. Library based on some descriptive or procedural languages (e.g. Fortran). Such an interface was the first. To implement library on any language it is necessary to rewrite all the code from beginning to end. Furthermore, it is difficult to study new nonstandard descriptive languages unknown to programming society and it is complicated to use the procedural languages, because CAD systems operate complex data structures.

2. Object-oriented library (e.g. C++). Coding complex procedures successfully such an interface solves the problem of data presentation. Object-oriented technique is the unified and language independent way of definition of CAD objects. Since object-oriented libraries don't concern the model of software system they can be reused only in the programming language framework.

3. Component library (e.g. COM). Describing the model of software system component approach provides the highest possibilities to reuse software modules. Such an interface is flexible because it conforms to object-oriented technique. Besides, it is language independent and able to integrate with any applications. Leading CAD systems (SolidWorks, AutoCAD) are componentware. For example, SolidWorks [2] is built on COM technique [3, 4] and provided with C++ library.

2.3 Mechanical Desktop

Mechanical Desktop R6 (MDT) provides the tools to operate with solid AutoCAD models and with the models imported from different solid modeling system. (MDT includes STEP, IGES and other translators) [5]. Importing 3D model, MDT scans it through and translates separate items into the parts. Some parts combined into one are considered as the assembly. Thus, import results in the tree of drawing components based on non-parametric solid. Basic MDT solids may be supplied with parametric solids: holes, facets, junctions, frames etc. All the components may be edited with help of AutoCAD and MDT tools. CAD systems have the tool to manipulate the separate components: to move, to rotate, to range to copy etc. MDT includes 3D Manipulator, which depicted on Fig. 1.

Mechanical Desktop provides:

- Improved solid modelling;
- NURBS-surface modelling;
- Refined assembly modelling;
- Automation and associative drawing;
- Data exchange with different CAD systems.

Besides, MDT may be reconfigured and reprogrammed and so there are many applications coupled with it, which provides altogether through-engineering from design to control programs. MDT is based on AutoCAD object model, implemented on COM technique (Fig. 2). All the objects may be grouped on two groups: environmental objects (Application, Document, MenuBar, ...) and geometric objects (3DFace, 3DPoly, 3DSolid, ...). Interconnecting with the first group programmer may customize user interface;

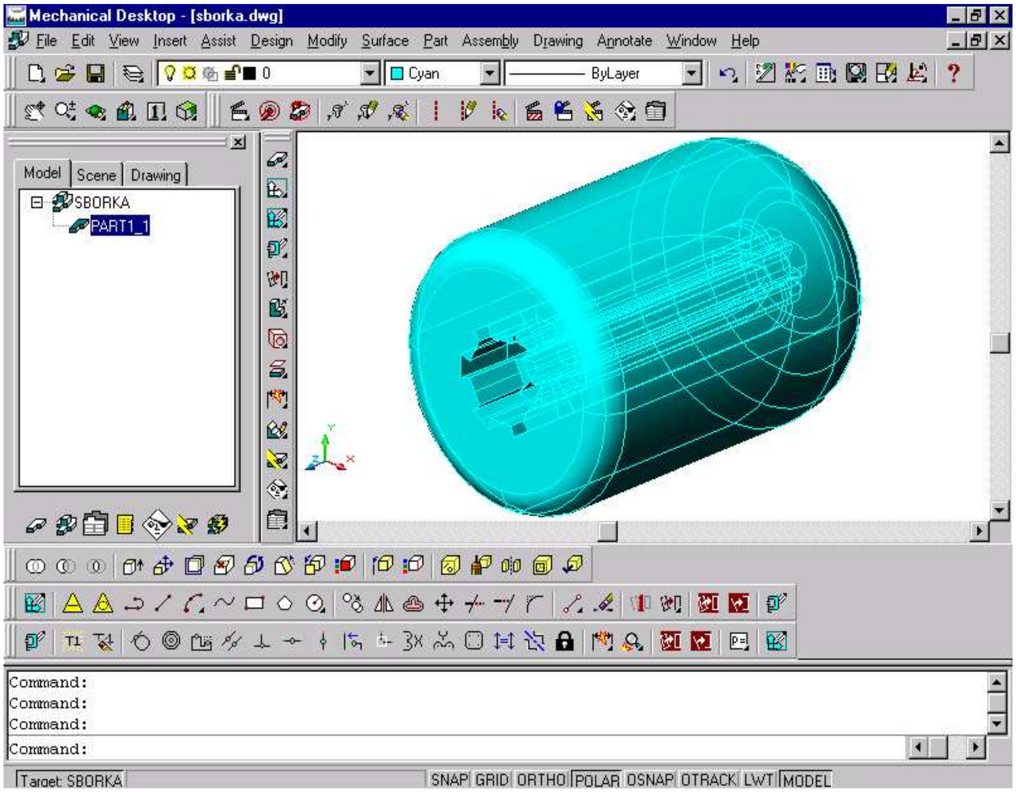

Figure 1: Mechanical Desktop Environment

the objects of the second one provides COM interface to ACIS geometric kernel. For more detail, see [6].

3 3D Mesh Generators

Mesh generators are designated to create 2D and 3D discrete models (meshes), which may be used in computation programs based on area discretization. The sequence of operations is as following: entering the geometry data, creation of the mesh, mesh refinement repeated as many as necessary to answer the limits imposed on mesh quality (maximum and minimum possible element angles, average element size, aspect rations etc). Mesh generator consists of the executive system and the geometric kernel.

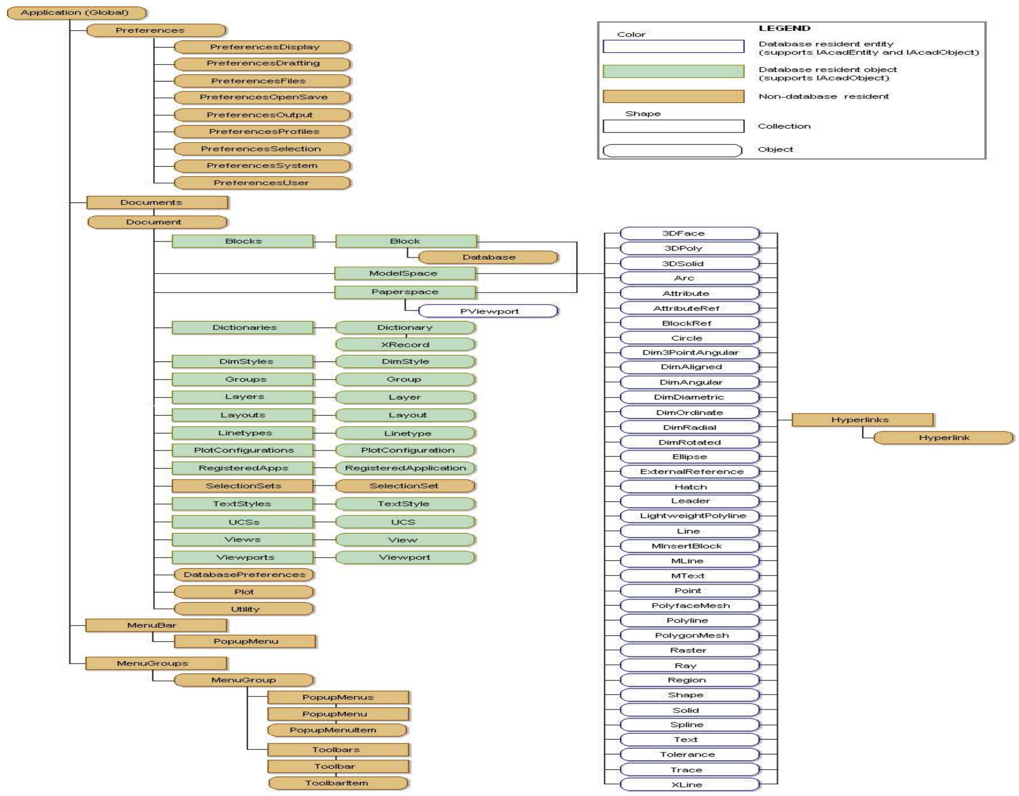

Figure 2: AutoCAD object model

An executive system provides the interface to operate with mesh generator in runtime: to set mesh quality parameters, to create and refine the meshes.

A geometric kernel is the integral part of mesh generators, but his main purpose is to completely describe the geometry and to take all the necessary data to perform a mesh generation algorithm.

3.1 The Generation of 3D Unstructured Meshes

Automatic mesh generation for complex two and three dimensional domains has recently become a topic of intensive research. It is imperative that automatic mesh generation tools be capable of generating quality finite element meshes. There must be a balance between resolution of the boundary and surface features and complexity of the problem. In addition, for problems with isotropic physics, element aspect ratio must be small to minimize linear system condition number and interpolation error. On the other hand, problems with anisotropic physics (for example, a shear layer in viscous fluid flow) require highly anisotropic elements for efficient solution. A further level of complication is that for some physical problems and applications, quadrilateral (2D) or hexahedral (3D) elements are preferred, even though filling space with high quality elements is easier using triangular (2D) or tetrahedral (3D) elements. For more detail, see [7, 8].

3.2 Programmer Interfaces of 3D Mesh Generators

There are many mesh generators directed towards the applications and implemented different algorithms of mesh creation. Reasoning from practice user decides what mesh generator meets the requirements. Now the interfaces of mesh generators are on the first and second development stages, so that the most suitable of them have an object-oriented interface and an object-oriented geometric kernel. Component approach helps to create unified interfaces encapsulated any mesh generation library. Universal generator component provides the connect to new generation library without any modifications of computation program. The work given realizes this approach. As an example freeware object-oriented library GRUMMP is used.

As for the object interfaces of geometric kernels, they have to include the total geometry data. For example, it is necessary to extend the interface of element. Not only does element include the set of nodes, but it is to be described by the wireframe (the set of edges) and surface (the set of faces).

3.3 GRUMMP

GRUMMP [9] is a set of libraries, written in C++, supporting unstructured mesh creation and modification in two and three dimensions, and a set of executables built on those libraries. The current publicly released version is capable of automatically generating triangular and tetrahedral meshes from polygonal and polyhedral input, respectively. GRUMMP determines appropriate mesh element sizes from the geometry and generates well-shaped elements, all with no user intervention. This library has the developed geometric kernel, written in C++ (Fig. 3). First, to guarantee that GRUMMP always generates high-quality meshes, algorithms by Ruppert (for two dimensions) and Shewchuk (for three dimensions) have been implemented, which

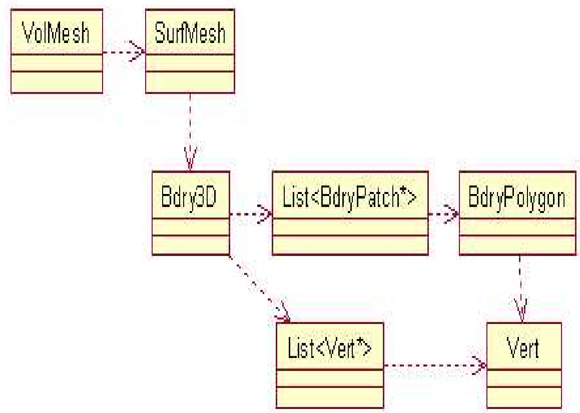

Figure 3: GRUMMP object model

produce meshes with provably well-shaped elements. These algorithms have also been modified to allow better user control of element size and rate of size variation.

Second, GRUMMP is being extended to allow user-defined definition of local element size; the main application for this extension is solution-based mesh refinement, which changes local mesh size based on a measure of solution error.

Real-world geometries are of course not all obediently polygonal and polyhedral. In practice, a software tool for industrial applications must generate high-quality meshes from clean CAD data. To address this problem, the GRUMMP implementations of Ruppert's and Shewchuk's algorithms will be generalized to allow (more or less) arbitrary input data.

The current release of GRUMMP is version 0.2.0. This version consists of executable programs:

- tetra generates three-dimensional tetrahedral isotropic unstructured meshes, using Shewchuk's algorithm modified for cell size and grading control [10];

- meshopt3d improves existing three-dimensional tetrahedral isotropic unstructured meshes, using techniques developed by Freitag and Ollivier-Gooch [11];

- coarsen3d produces a tetrahedral unstructured mesh with approximately twice the local length scale of the input mesh, including directional anisotropic coarsening of quasi-structured parts of the fine mesh.

When the simplicial mesh generation code is stable and robust, attention will turn to generation of mixed-element meshes. For this case, point placement strategies are especially important to ensure that tetrahedra can be combined to form high-quality hexahedra, prisms and pyramids. These element types will be formed in a post-processing step.

4 The Integration of Mechanical Desktop and GRUMMP

The integration of Mechanical Desktop and GRUMMP puts into practice some ideas mentioned above

1. Switch from formatted files to programmer interface to exchange geometric data.

2. Development of mesh producing interface to encapsulate any mesh generator.

3. Continuous process: design – problem statement – computation.

To discuss the details of the integration, object and functional models of the integrated system are reviewed.

4.1 The Object Model of the Integrated System

The integrated system consists of following subsystems: MDT, unstructured 3D mesh generator GRUUMP, integrator and computation module. The integrator consists of COM-object encapsulated C++ library and VBA-project connected with MDT. COM-object is the prototype of universal mesh generator component. In the future it is supposed to extend depending on what mesh generator is to be integrated.

4.2 The Functional Model of the Integrated System

Integrator's objects interact with each other by C++ and COM technique. The interactions are described in the following way (Fig. 4).

1. MDT and VBA-project are activated simultaneously. When drawing (Fig. 5) is being opened VBA-project is extending MDT interface and inserting new menu items to configure and run the mesh generator.

2. Mesh generator parameters are set.

3. In MDT environment the new document is opened to contain 3D mesh that will be generated.

4. COM-object encapsulated GRUMMP library takes the pointers to the geometric model. Then the MDT geometric model mapped onto the generator geometric model. As a matter of fact it is converting between two geometric formats. This case is possible because the two geometrical kernels have equal power. COM technology and object-oriented approach manage to avoid hard programming when data are converted through the formatted files.

5. On the design mapped into the set of C++ objects mesh generator produces the mesh.

6. COM-object takes the pointer to container drawing. Then the mesh geometric model formed by mesh generator is mapped to MDT form and copied into the container draft to visualize (Fig. 6).

7. Numeric parameters are assigned to elements on the 3D mesh. Thus geometric mesh becomes computational grid.

8. Computational grid is sent at the input of computation module.

5 Conclusion

The integration of CAD-systems and mesh generators is the productive way to get the data to compute. This is of major practical importance to develop complex software systems. This approach is to be used in the context of development of the high-performance finite-element application framework.

Acknowledgments This investigation was accomplished with support of Russia Foundation of Fundamental Investigations (projects 99-07-90455, 02 -07-90265).

References

1. Mathematics and CAD. – M.: Mir, 1989, 264 p.
2. SolidWorks 2000 API.
3. Brockschmidt K. Inside OLE. 2-nd ed. – M.: Microsoft Press, 1995.
4. Rogerson D. Inside COM. 2-nd ed. – M.:Russian redaction, 2000, 400p.
5. Nagradova M. AutoCAD. Designer guide. – M.: Prometey, 1991.
6. AutoCAD 2002 Help.
7. Preparata F., Sheynos M. Computation geometry. Introduction. – M.: Mir, 1989, 478 p.
8. Alies M.U., Kopyssov S.P., Varnavski A.I., Novikov A.K. The Construction of Voronoi's diagrams and Delone triangulation in two and three dimensions. – Preprint of IAM UB RAS, 1996, 39 p.
9. Ollivier-Gooch C. GRUMMP Version 0.2.0 User's Guide // Department of Mechanical Engineering University of British Columbia, 2001, 26 p.
10 . Freitag L. A., Ollivier-Gooch C. Tetrahedral Mesh Improvement Using Swapping and Smoothing // Int. J. for Numerical Methods in Engineering, 40, 1997, pp. 3979-4002.
11. Ollivier-Gooch C., Boivin C. Guaranteed-Quality Simplicial Mesh Generation with Cell Size and Grading Control // To appear in Engineering with Computers, 2001.

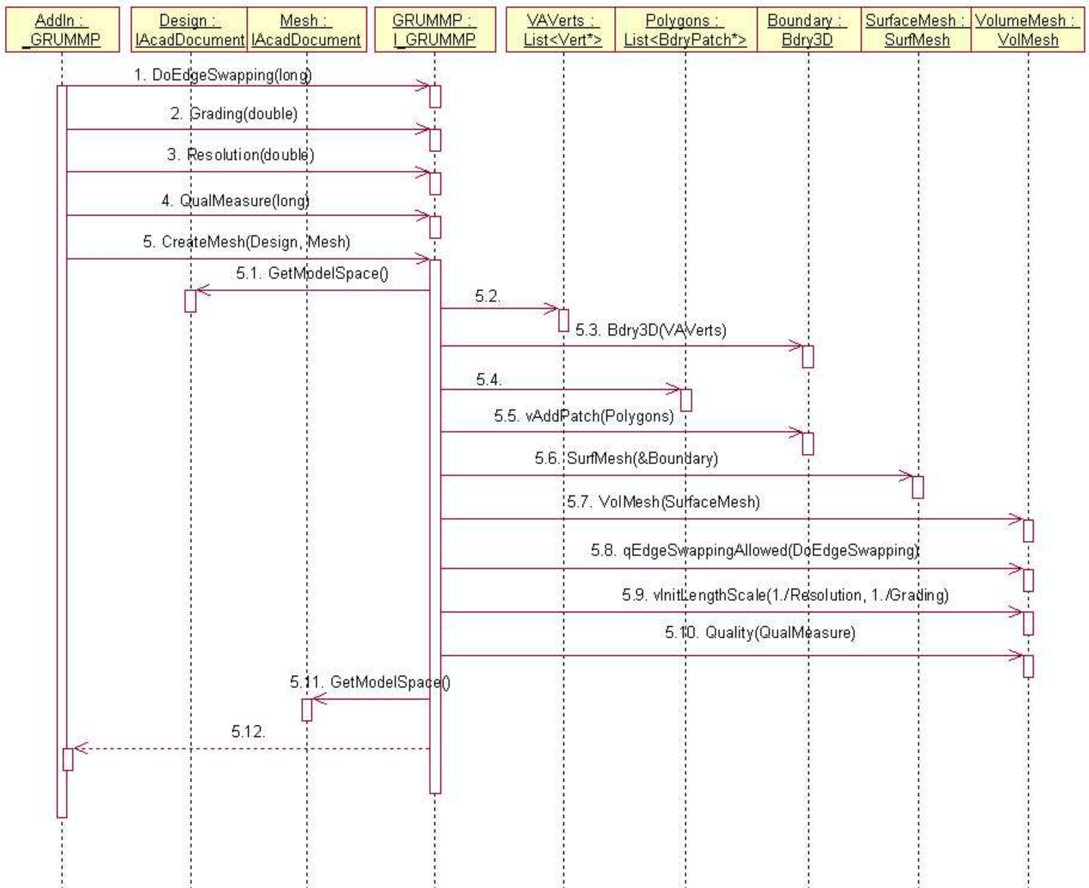

Figure 4: Mesh generating

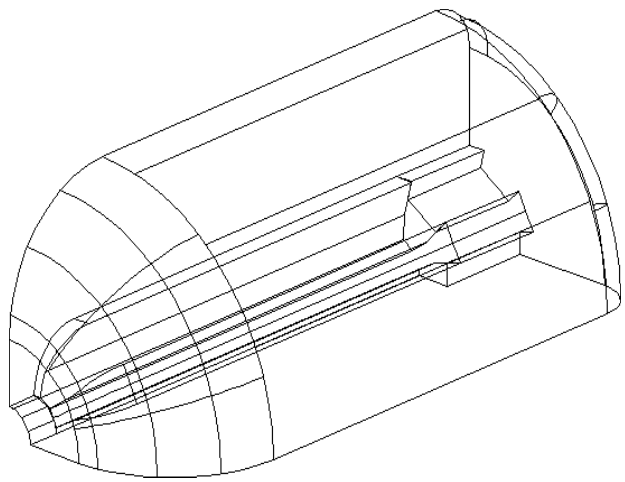

Figure 5: The surface model

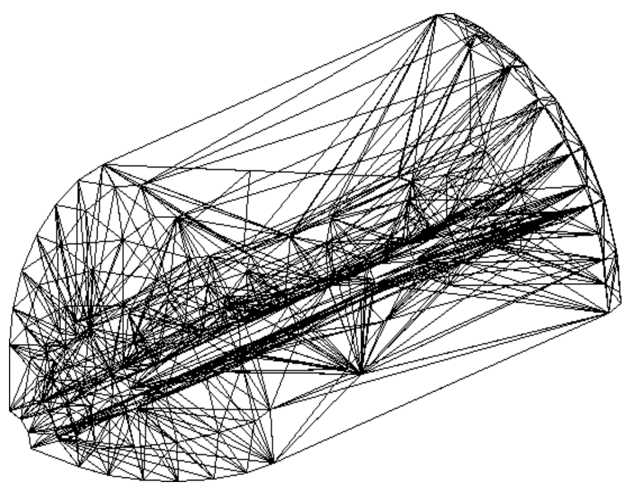

Figure 6: Unstructured tetrahedral mesh