Automation techniques for construction of solid models in CAD

Ahamed, S. S.

Publisher's version / Version de l'éditeur:

Flexible Automation and Intelligent Manufacturing (FAIM) 2004 [Proceedings], pp. 1301-1307, 2004-07-01

Automation techniques for construction of solid models in CAD

IMTI-XP-258

Ahamed, S.S.

A version of this document is published in / Une version de ce document se trouve dans:
Flexible Automation and Intelligent Manufacturing (FAIM) 2004, Toronto, Ontario, July 12, 2004, pp. 1301-1307

Automation Techniques for Construction of Solid Models in CAD

Syed Shafee Ahamed*

Integrated Manufacturing Technologies Institute
National Research Council of Canada
London, Ontario, N6G 4X8, Canada

ABSTRACT

CAD systems offer many different methodologies to construct design models. 3D solid models are becoming more popular in the engineering design. Construction of 3D solid models comprises of two major techniques namely, Boundary Representations (B-rep) and Constructive Solid Geometry (CSG). The geometry required for these techniques is derived from wireframe and surface modeling methods.

CAD automation involves generating design models faster. Scripts and macros are used inside CAD systems to quickly aid construction of models. Skills and knowledge of the user becomes crucial in developing the scripts and macros. Also, because of the inherent nature of the scripts, user interactions are limited.

Presently, design models generated on PC based CAD systems running on Windows operating system can be treated as objects and can be linked with other desktop applications. Conversely, the CAD objects can be created using data derived from other applications. A system using Application Programming Interface to construct solid models interactively using array of point data derived from different application is developed. The current system is implemented in SolidWorks® CAD software environment.

1. INTRODUCTION

Presently, CAD has fully replaced the manual drafting practices to represent the design models. New techniques such as 3D modeling, model rendering and model simulations are greatly contributing towards better and improved product designs. With the advances in the CAD systems and supporting computing environments the quest for knowledge-driven, faster and better product design is increasing.

Reducing the design cycle time, eliminate design errors and increase productivity are the three important factors in design engineering process. Presently, the design models are also expected to be consistent across platforms (interoperability) and designers would like to interact with the models (interactivity). The material presented in this paper and the prototype example describes ways in which design models can be quickly generated interactively using automation techniques via ActiveX controls. SolidWorks® CAD Application Programming Interface (API) is used to invoke the required methods for construction of solid model objects.

2. LITERATURE REVIEW

In the past several techniques have been used to automate the process of design. Specifically, many geometric methods are used to generate solid models. One of the earliest works in generation of solid models was by Anderson and Chang [1]. They used limited set of features to construct a CSG like shape model to produce process plans for machining operations. Bloomenthal, et al [2], developed a "Sketch-N-Make" system which enabled individuals to create dimensional designs quickly for prismatic parts. The system also helped users to develop a smooth and rapid pathway from concept to detailed design for limited domain of machined and plastic parts.

* Corresponding author: Tel.:(519) 430-7087; Fax: (519) 430-7032; E-mail: shafee.ahamed@nrc.ca

In the recent years many advances in the graphic technologies used in engineering design, especially in modeling and display areas have emerged. A brief overview of these modeling technologies is described by Ault [3]. Apart from the evolution of the CAD systems and modeling techniques, the work also focused on describing the associations of solid modeling with the technology of using Computer-aided design to define the shape of the geometric structures. In the same year, Miller [4] from Purdue university described new technologies for engineering graphics. The research highlighted the historical design representation method to engineering computer description of part geometry. It also discussed about the emerging engineering graphic techniques such as visualization, knowledge-based sketches and associated hardware tools.

Advances in computer hardware and software technology have reduced the gap between low-end and high-end CAD systems. Williams [5] has highlighted the effect of these on the design domain and discussed about the Design-centric vs the Process-centric CAD operations and their interoperability issues. Later many schemas were developed to support both Design-centric and Process-centric CAD data and information. Other methods for exchange of parametric information is reported as in [6] by Choi, et al. Their method is intended to provide exchange capabilities for parametric information including design intent via macro files. They also listed the drawbacks and limitations of some of the present data exchange standards.

In the present design environments, CAD engineers are responsible for designing, drafting, analysis including digital prototyping. Thus, automation in design process is important and greatly reduces the product development time and increases the quality of design. Barr, et al [7] describes the integration of digital design techniques and graphic processes to aid the process of design cycle.

Many commercial tools and technologies are now available to assist in integration and automation of design process among different applications. Winters [8] describes the methodology for integration of Design-centric CAD system with Windows-based application programs. Finally, more research findings in areas of automation technologies for CAD design using OLE techniques are compiled by Sendler as in [9].

3. BACKGROUND

CAD systems are a collection of software tools and methodologies to construct, modify, display and store design shapes. These CAD systems have to cater a wide variety of applications such as engineering, architecture, etc. Also, these applications govern the design process. Thus, the domain of design automation has to encompass both, application area and the CAD system. Generally, the process of automation can be classified into two broad categories namely, customization of the CAD systems and programming interfaces.

The process of customization involves re-arranging and grouping of CAD commands and functions in a way the designer will intend to use it. Generally, the frequency of usage of commands, or commands used for a particular task is the criteria for customization. Most of the commercial CAD systems provide toolbar, menu and screen customization. Customization leads to quicken the process of design.

Programming interfaces are used where repetitive tasks inside the CAD systems are required. Generally, these interfaces include scripts or macros. Scripts are ASCII CAD commands stored in a file and then run to mimic as if they were input from the keyboard or mouse. Scripts and macros are proprietary in syntax and are very sensitive. Figure 1 shows a script file to generate a solid shape as shown. The syntax and names used in this file are proprietary to AutoDesk® systems.

```
box 0,0,0, 1 10 5 2
wedge 0,0,2 1 2 5 10
(setq object1 (entlast))
ucs 3p 2,5,2 0,5,12 2,0,2
circle 2.8,2,5-1.4 1.0
extrude 1 3 0
(setq object2 (entlast))
subtract !object1 object2

 vpoint 1,1,1
```

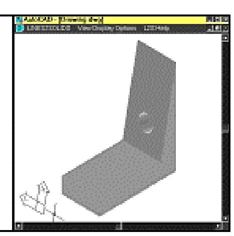


Figure 1: Sample script file in AutoCAD CAD system

Additional features can be added to these scripts and macros to achieve system customization. These scripting methods are extremely fast and can generate complex models in seconds provided the supplied script data is appropriate and free from syntax errors. The major drawback of scripting or macros being, formats and structure are proprietary, cumbersome and has limited capabilities in logic control and variable definitions. In order to overcome the limitations of providing logic controls and data to be represented as variables, many CAD systems came up with their own proprietary languages. These languages were similar to standard programming languages such as FORTRAN, PASCAL and C. These proprietary languages allowed the user to create parametric models. Presently, interfaces can be written in standard programming languages like C, C++, JAVA and Visual BASIC. Object calls to create, manipulate and store can be made. Once these modules are compiled they can be customized inside the CAD system as if they were built-in. This greatly enhances the interface capabilities.

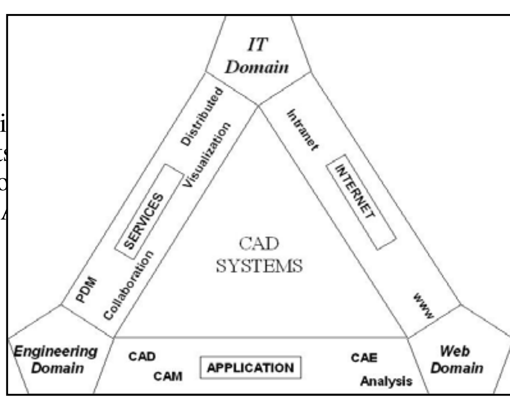
Programming interfaces are also used where different sets of data are passed onto the CAD system from other applications to quickly generate CAD models or extract geometric data from design models for downstream applications such as analysis and manufacture.

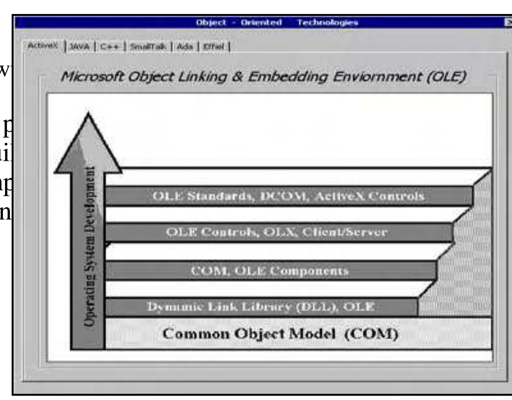
Depending on the CAD environment, application programmers have been using proprietary, standard programming languages or programming interfaces of window-based applications to directly generate scripts and macros suitable for a given CAD system. Linking CAD systems to information contained in other applications to automate the construction of actual design models have resulted in valuable cycle time reduction and improved quality. These techniques have proved to be very effective especially, to create variety of design configurations

31. OPERATING SYSTEM

Until recently, CAD systems have been used as a customized geometry kernel or efficient geometry and display engine to perform the process of design and drafting. Presently, the whole concept of usage of CAD is getting blended with the desktop technology. Left side of Figure 2 shows the domain in which the CAD systems are operating presently.

Recent developments in the Operating System (OS) of desktop environments are allowing the end user to do things much faster, better, able to share and communicate. Currently they offer capabilities for interaction and interoperability. These functions can be embedded in the design automation to create and interact with the CAD environment. Right side of Figure 2 shows the growth of ActiveX technology within the Microsoft operating environment.





grow

ActiveX i … gies p … an
access object … e bui … gy
is the Compo … comp … un
anywhere in … nown

3.2. MODELING TECHNIQUE

Modeling techniques in design have evolved from the early stages of 2D drafting to advance solid modeling techniques. The most common modeling techniques presently available are shown in Figure 3.

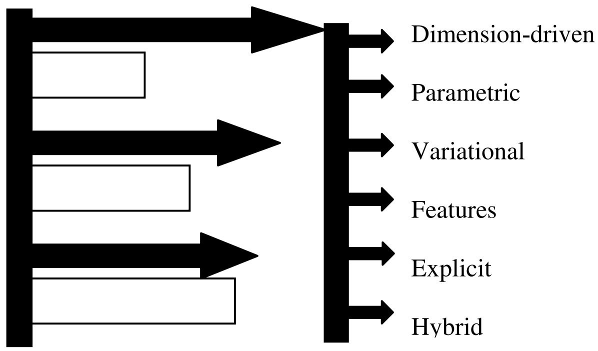


Figure 3: Modeling techniques in CAD

Solid models can be constructed in two fundamental ways, Boundary Representation (B-Rep) and Constructive Solid Geometry (CSG). B-rep uses construction methods wherein, solids are constructed by extrusion/sweep of closed bounded curves or regions. CSG does model building using standard primitives and later boolean operations are applied to edit the shape of the solid. Generally, in mechanical engineering design process a combination of these methods is used to construct the design models. The solid modeling techniques used in this work are explained below:

3.2.1 EXTRUSION METHOD

Constructing solids using extrusion method consists of defining a closed/bounded curve called the contour or region. This contour curve may consist of many edge element connected to each other in an ordered loop. Later the contour curve is extruded in a direction defined by a direction vector. Figure 4 illustrates the extrusion method.

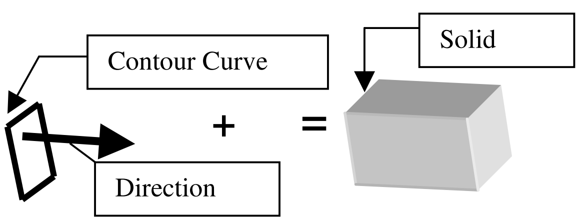


Figure 4: Extrusion method

3.2.2 BOUNDED VOLUME METHOD

Bounded volume method of constructing a solid consists of defining bounded curves to build a surface. Later these individual surfaces are stitched together to form a single surface. The solid is generated by the volume enclosed by the bounded surface. The process of constructing a solid by bounded volume method is illustrated in Figure 5.

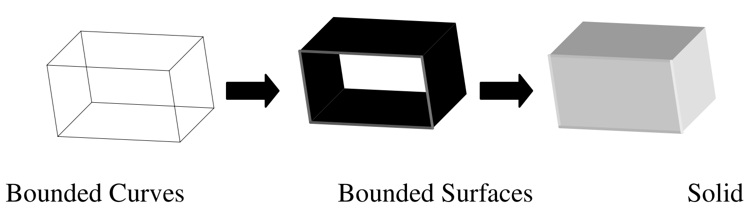


Figure 5: Bounded volume method

4. AUTOMATION IN DESIGN MODELING

In the CAD/CAM/CAE environments integrating applications is typically done via exporting and importing CAD data in neutral formats. AutoDesk® proprietary format DXF has become default standards in exchanging 2D information. Initial Graphic Exchange Standards (IGES) is commonly used in exchanging the surface data to downstream applications. Solid Geometry data are typically translated by STandard for Exchange of Product data (STEP).

Presently, it is realized that interaction with the model environment is important to improve the design automation process. CAD systems are providing interfaces via API's, which enables the user to interact with the model environment. Also, with the increasing need for sharing data across applications from various disciplines to provide solutions for real-world problems, attempts are made to adopt open standards that facilitate data sharing. Microsoft's Object Linking and Embedding (OLE) technology provides solutions for inter-operability between applications.

OLE for Design and Modeling (OLE for D&M) is a logical extension of Microsoft OLE technology from office applications to 3D CAD/CAM and CAE environments. It aims to directly translate, construct and visualize 3D CAD objects without converting it into neutral formats. OLE for D&M supports the concepts of Microsoft client/server and container functions to exchange 3D data.

The advantage of OLE in design is the freedom and flexibility it gives to construct complex models. The user has to invoke the object methods and supply the parameters required. Interactivity can be build by functions provided by the windows operating system. These technologies are allowing users to build applications with objects being shared and controlled across different applications on desktop platforms.

4.1. ACTIVE X AUTOMATION INTERFACE

ActiveX Automation Interface (AAI) for CAD consists of CAD applications running simultaneously and providing programming control over CAD objects to construct, inquire and edit objects across other applications in a bi-directional way. AAI provides methods to manipulate objects within or outside the applications by exposing the objects to the "outside world" which can be accessed. A typical AAI environment consists of an open architecture CAD system wherein the objects can be accessed, an ActiveX protocol to enable communication and a supporting programming language to build logics and controls.

ActiveX automation in CAD systems also allows users to customize and integrate using computer programming languages that support the AAI such as C++, Delphi, Visual BASIC (VB) and JAVA. Currently VB is the most common application development platform in CAD, called as Visual BASIC Applications (VBA). VBA is now becoming an integral part of the Windows-based desktop CAD systems. Creating VBA applications for automation requires invoking of CAD models, selection of CAD entities, creation/manipulation of entities and displaying the results. Since VBA is tightly integrated with windows operating system environment it has access to the Windows functions, file system and the computing hardware. In other words, building applications in CAD/CAM/CAE using ActiveX automation is achieved between server controller, an application that controls the server application (CAD system) and VBA as an Automation Controller.

In the process of ActiveX automation, CAD entities, objects and functions have a *method* and *properties* associated with it. Through the programming interface these properties can be accessed. Generally a *property* refers to an attribute of an object and *methods* are functions that perform an action on an object.

Automation in design is achieved by a combination of accessing the right method, properties and using them in a logical sequence. Generally, the code responsible for automation consists of declarations, data, logics, initializing, assignment and control statements. The window for interaction with the user and the CAD system can be achieved by creating a Graphic User Interface (GUI). The GUI will serve as a window for exchange and flow of data between applications.

5. SUMMARY OF WORK AND PROTOTYPE EXAMPLE

A prototype system has been developed to automate the process of building solid model from an array of point data in SolidWorks® CAD environment. A GUI is provided to interact with the CAD system and to aid the process of construction of solids. The work is divided into three major modules. The first module constructs a 3D sketch wireframe model from the data set. The second and the third modules provide two different construction methods to build solid geometry. Figure 6 illustrates the flow and the sub-modules developed to accomplish solid models from data set. Figure 7 show the screenshots of the generated wireframe model, final solid model and the developed user interface inside SolidWorks® environment.

The first construction method uses a closed contour or closed loop of edge elements and a direction vector to define a solid element. The sub-module "EntitySelection" will allow the user to pick sketch/wire entities created by the sketch generator module from the graphic screen or the feature manager window. The "CheckEdges" sub-module will determine the continuity among adjacent edges to make sure that there are no gaps in between them. The "BuildLoop" sub-module will orient these edges in a counter-clock wise direction and will produce a single loop or contour. As a final step this contour is extruded along the edge direction defined by the user by the "LoopExtrude" module to construct a solid model.

The second method uses bounded surface information to build solid models. The procedure is similar to the previous method up to building individual loops. Later each loop is used by the "BuildSurface" sub-module to construct individual surfaces. The edges in each loop are used as bounding curves or bounding edges for the surface construction. This procedure is repeated to define all other bounding surfaces. Once all the surfaces are defined then the "StitchSurface" sub-module will stitch these surfaces together to form a single bounded surface. Finally the "BoundedSurface" sub-module will construct a solid by using this single bounded surface. Few error routines have also been incorporated to help the user with appropriate messages along the process of construction of solids.

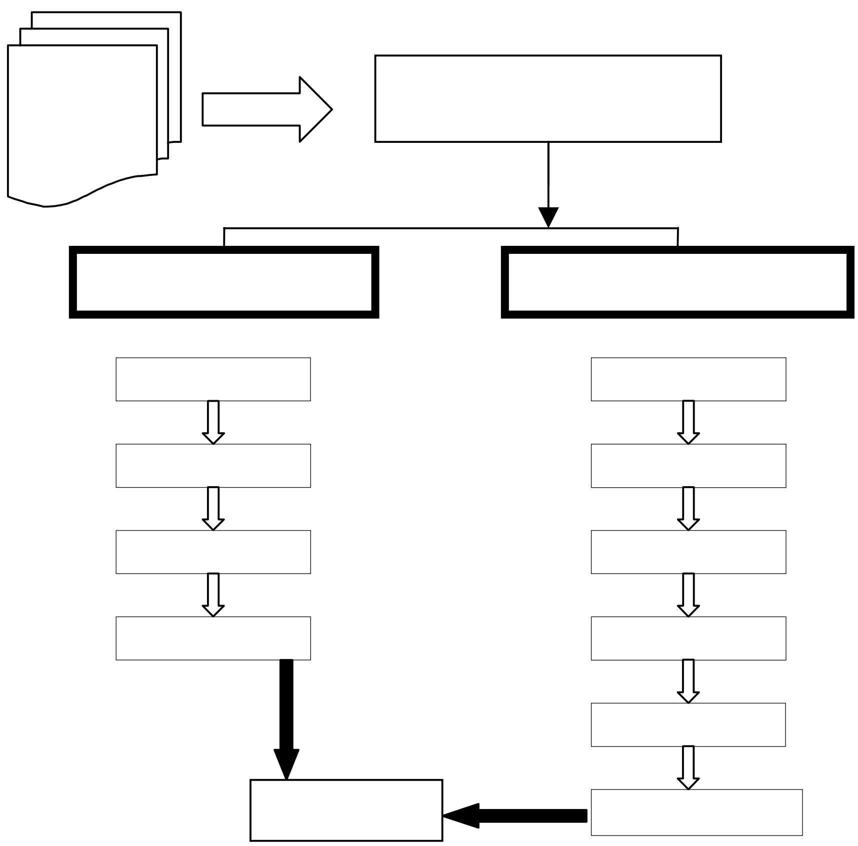


Figure 6: Flow and modules of the developed system

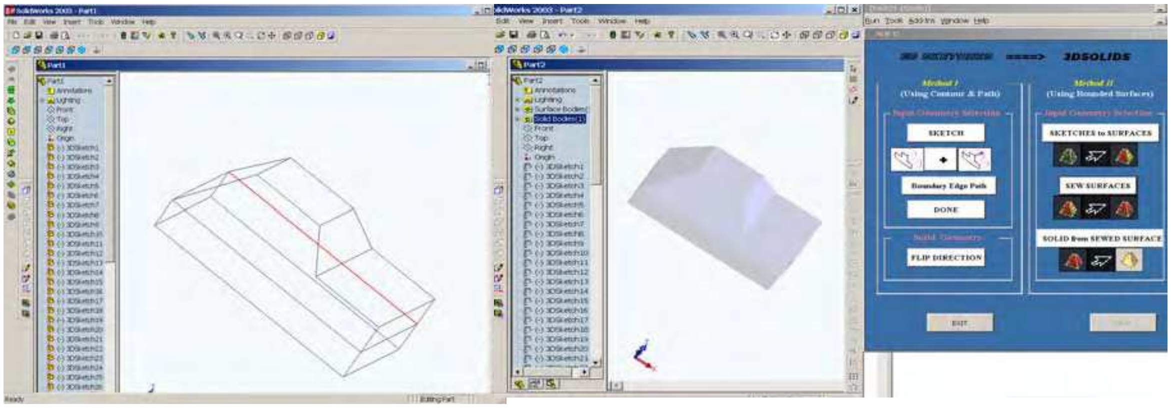


Figure7: Generated wireframe and constructed 3D solid model with user interface

6. CONCLUSIONS

The prototype system developed provides a smooth, error-free and quick way of building solid models from an array of point data. It demonstrates the integration of user developed modules with the CAD system. This system also allows the user to interact and exchange objects within the Microsoft windows environment. The advantages of this system have been achieved by integrating application programs and construction methods by a user friendly interface. Users with no skills in CAD operations or CAD environments are able to use this system.

The system in its present form is serving as a vital tool for quickly constructing 3D solid models in projects involving reverse engineering applications.

ACKNOWLEDGEMENTS

Author is grateful to A. Pardasani, Senior Researcher, at Integrated Manufacturing Technologies Institute for his feedback and suggestions during the actual work and writing this paper.

REFERENCES

[1] *D.C.Anderson and T.C.Chang: "Geometric Reasoning in Feature-based Design and Process Planning", Computers & Graphics, Vol. 14, No 2, pp 225-235, 1990.*

[2] *M. Bloomenthal, R. Zeleznik, R. Fish, L. Holden, A. Forsberg, R. Riesenfeld, M. Cutts, S. Drake, H. Fuch and E.Cohen: "Sketch-N-Make: Automated Machining of CAD Sketches", Proceedings of DETC'98, 1998 ASME Design Engineering Technical Conferences, pp 1-11, September, 1998.*

[3] *H. K. Ault: "3-D Geometric Modeling for the 21st Century", The Engineering Design Graphics Journal, Vol. 63, No. 2, pp 33-42, 1999.*

[4] *C. L. Miller: "New Technologies for Engineering Graphics", The Engineering Design Graphics Journal, Vol. 63, No. 2, pp 50-57, 1999.*

[5] *A. William: "Design-Centric vs Process-Centric CAD", CADANCE, Vol. 14, No. 5, pp 100, 1999.*

[6] G. Choi, D. Mun and S. Han: "Exchange of CAD Part Models Based on the Macro-Parametric Approach", International Journal of CAD/CAM, Vol. 2, No. 1, pp 13-21, 2002.

[7] R. E. Barr, T. J. Krueger and T. A. Aanstoos: "The New Digital Engineering Design and Graphics Process", The Engineering Design Graphics Journal, Vol. 66, No. 3, pp 6-11, 2002.

[8] J. Winters: "Customizing AutoCAD with Microsoft Excel", VB CAD, 1998.

[9] U. Sendler: "CAD Office Integration", Springer, 1995.