

THE CLEMSON INTELLIGENT DESIGN ENVIRONMENT FOR STEREOLITHOGRAPHY – CIDES 2.0

Jill S. Kirschman
Industrial Engineering Department
jkirsch@ces.clemson.edu

Charles F. Kirschman
BPM Technology Inc.

Georges M. Fadel
Mechanical Engineering Department
gfadel@ces.clemson.edu

Joel S. Greenstein
Industrial Engineering Department
jsg@ces.clemson.edu

Laboratory to Advance Industrial Prototyping
Center for Advanced Manufacturing
Clemson University
Clemson, SC 29634 USA

ABSTRACT

There are a large number of commercial Rapid Prototyping (RP) devices available today. All of these machines begin with a Computer-Aided Design (CAD) model, which is tessellated, sliced and then built layer-by-layer on the RP device. All of these operations, except the actual building of the part, are completed on a computer. Therefore, many improvements to the RP processes can be achieved through software, without affecting the RP devices or the warranties on them. This has led to the development of a front-end software product to support the task of preparing the part to be built. The Clemson Intelligent Design Environment for Stereolithography (CIDES) is a user-centered interface between the CAD system and RP systems, primarily the Stereolithography Apparatus (SLA).

CIDES 2.0 is designed to provide a variety of tools which are valuable to the users of RP systems, including the ability to view and modify tessellated (STL) files, generate supports, and slice STL files into layer (SLI) files for use on an SLA. It also provides the ability to view SLI and merged (V) files. Furthermore, CIDES offers additional translation capabilities that make it valuable for other RP processes. The package has proven useful in the Laboratory to Advance Industrial Prototyping (LAIP) at Clemson University. CIDES 2.0 is a new X Windows-based release based on the original version of CIDES with many additional features. A new Human-Computer Interface is the major improvement to this release.

INTRODUCTION

Rapid Prototyping (RP) is the name for a number of processes that generate three-dimensional objects directly from a Computer-Aided Design (CAD) solid model. Regardless of which process is used, the procedure to create these objects is to generate the CAD model, tessellate it, slice it, and finally build it on the RP device. All of these steps, except the actual building of the part, are completed on a computer. This was the motivation for the development

of a front-end software package to support the task of preparing a part to be built (Kirschman et al., 1991A). The Clemson Intelligent Design Environment for Stereolithography (CIDES) is fundamentally a user-centered interface between a CAD system and the Stereolithography Apparatus (SLA).

Rapid Prototyping is an active research area. Many researchers have custom-designed devices to conduct their experiments; however, much research is done using commercially available machines. Researchers can achieve significant changes to the parts produced on these machines without voiding the manufacturer's warranty by making changes to input files, or by interacting with the layer files that will be used to build a part. CIDES was developed for this purpose; it includes a number of features that have enabled Clemson University to enhance the capabilities of the RP systems that it has. CIDES provides input, output, and viewing capabilities for several common file formats. It also enables input file modification, support generation, slicing, and file format translation. A number of research activities have been based on the CIDES platform, including support structure generation (Kirschman et al., 1991B), optimal orientation (Frank and Fadel, 1994), NURBS-based slicing (Vuyyuru et al., 1994), adaptive slicing (Tata, 1995), and optimal placement of parts (Wodziak et al., 1994). Currently CIDES is closely coupled with an SLA because that was the first available RP device, and the only one at Clemson University for several years. However, the code is being expanded to work with other RP systems, including the selective laser sintering testbed at Clemson University (Terala, 1996).

The original version of CIDES (version 1.0) was written in 1991. CIDES 1.0 runs on SunOS 4.3 or earlier. However, Clemson University has nearly completed the process of upgrading from this operating system to Solaris on all Sun workstations. Although the kernel of functionality is very portable, the CIDES 1.0 graphical user interface (GUI) and graphics engine were developed using the Hierarchical Objected Oriented Programming System (HOOPS) from Ithaca Software. HOOPS was an excellent choice for a graphics system in 1991, when there were few GUI standards for workstations. It is highly portable with versions for many platforms, and very powerful for display. However, HOOPS has two shortcomings. First, HOOPS is designed for graphics and object display; it does not offer the GUI-building capabilities of a typical fourth generation language or a GUI builder. Second, Ithaca Software imposes royalties for the distribution of HOOPS-based software. As a result, CIDES 1.0 can not be freely distributed. This is the primary reason that CIDES has not found its way outside Clemson University, despite heavy use in the Laboratory to Advance Industrial Prototyping (LAIP).

Another problem with the original version of CIDES is that it was designed for research purposes rather than general use. Very little attention was devoted to the user interface during the development of the product. Gould (1988) proposes that four principles of system of design be followed if one is to design a usable computer system. First, there should be an early and continual focus on the users. There should be direct contact with the users so as to understand their job and what they need to do their work better. Second, there should be integrated design. All aspects of usability should develop in parallel and under one focus or one person. Third, there should be early and continual user testing. To ensure that a design usable by and useful to its users, the system must be tested by its intended users. Finally, the design process must be

iterative. The system should be modified based on the results of user testing. Once the modifications are complete, user testing should be repeated. This refine-and-test cycle is repeated until testing demonstrates that the product satisfies its usability specifications. The methodology used to redesign the CIDES interface applied these four principles.

CIDES 2.0 was developed to achieve greater product accessibility. A new development platform, SGI workstation, was chosen and a new GUI was designed for the system. Several of the underlying data structures were revamped, and code was modularized to separate the functional kernel from the interface. This paper describes the redesign of the software from the top down and discusses the capabilities that are now available in the product.

METHODOLOGY

A four phase structured methodology was used to redesign the CIDES interface (Ulrich and Eppinger, 1995). This methodology was chosen because it involves users in the design process, is user-centered and is self documenting. The first phase of the process begins with the development of a mission statement. For the CIDES redesign project, the mission statement was:

Redesign the CIDES interface to make it more usable in and beyond the Laboratory to Advance Industrial Prototyping (LAIP) at Clemson University.

Once the scope of the project was defined, representative users were identified based on several different criteria, including familiarity with CIDES, exposure to CIDES and similar packages, and frequency of use. A thirty minute semi-structured interview was conducted with five of the nine users. The remaining four users were interviewed via email because of their geographical location or scheduling difficulties. Upon completion of these interviews, each user statement was translated into a user need statement. To determine the relative importance of each need, the users were then asked to rate each need statement using a five point scale.

The second phase of the design process translated the list of user needs into performance measures which were both testable and measurable. Some needs were not easily translated into quantifiable metrics. In such instances, metrics were created based on the user's subjective evaluations of the interactive system. Next, products similar to CIDES were identified and used as competitive benchmarks. The three products benchmarked were: the original version of CIDES, Partman (3D Systems) and Surfacer (Imageware). These products were benchmarked in terms of all the product performance metrics. The information obtained from the interviews with users and the benchmarking process was used to establish a set of product specifications.

Phase three consisted of concept generation and concept selection. It began with each design team member individually generating ideas. Then the design team brainstormed together to create additional concepts and refine the initial ones. This process was repeated several times. The concept selection process was also iterative. First, the team selected the user needs and product specifications most able to differentiate among the concepts and prepared a selection matrix. The selection criteria were then placed in the rows of the matrix in descending order of importance. Next, the team chose one concept as a reference, and all other concepts were rated

on a +, 0, - scale against this reference on each criteria. When there was no difference, or if doubt existed as to whether a concept was better or worse than the reference, a score of “0” was given. The concepts were ranked based on the number of +’s, 0’s and -’s assigned to them. The decision to carry a concept forward was based upon team consensus, taking into consideration the rank, strengths and weaknesses of the concept. As a result, the team decided that a concept employing a Toolbar, Menu and Hot Keys would be carried on to the detail design and development phase.

In the final phase, the selected concept was developed and tested. Employing a process similar to that used by IBM to develop the 1984 Olympic Message System (Gould et al., 1987), the team obtained feedback regarding the effectiveness of the interactive system being developed early and repeatedly. Initially the team used paper-and-pencil prototypes to test user reactions to the proposed design. Once the organization of tasks within the interface was completed the team began developing focused software prototypes. When the majority of the tasks were coded, the team moved to a comprehensive prototype so that users could identify any remaining design flaws before the overall design concept was frozen for detail design. Once detail design was complete, the interface was tested and refined using real tasks. Testing and refinement was an iterative process that began with the prototype developed in the detail design phase. Initial testing focused primarily on the layout of the interface as well as the labels used. This was followed by four iterations of additional testing and refinement. During each of the four iterations, as many of the users as practical were included in the product testing. After the four iterations of testing and refinement, the interface appeared to be ready for final evaluation against the product specifications. The final version of the interface met or exceeded every product specification when it was tested in the Laboratory to Advance Industrial Prototyping (LAIP).

STRUCTURE

CIDES 2.0 was redesigned to enhance portability and simplify improvements to the user interface. All of the core functionality of CIDES, including input/output, part manipulation, slicing, and supporting has been combined into a set of library routines that are called from the GUI (Figure 1). This separation enables the kernel, which is written in ANSI C, to be ported to different platforms more easily. Once the kernel is running, any interface from a simple text menu to a full graphical display or even a virtual reality helmet can be used on top of the kernel to provide access to the routines. The library contains routines to carry out all of the non-graphic functions capabilities discussed in the next section.

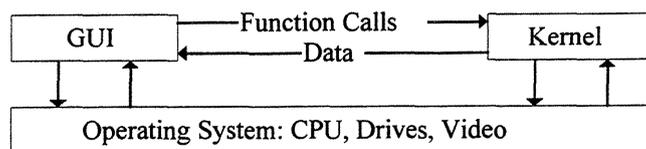


Figure 1. Kernel Structure of CIDES.

This library approach enhances the utility of the software for research. While changes to the kernel must be more stringently tested, researchers can use the basic capabilities of input/output

and slicing without the risk of damaging the base code. Thus, additional user functionality can be added through software modules by anyone with a working knowledge of C.

CAPABILITIES

Since CIDES is a research testbed, its capabilities may change rapidly. Its current capabilities can be broken into three categories: Display, Modification and manipulation, and Utilities. Each of these categories will be discussed in detail.

The first function of CIDES is to display files of various formats. Included in this must be the ability to read these files; currently STL, SLI and V files are supported. STL files are industry standard tessellated files used as input to all RP processes. CIDES supports reading and writing both ASCII and binary files that follow the STL file specification. Because some CAD packages generate non-standard STL files, CIDES is capable of handling several common anomalies, including non-standard headers or lines, incorrect case or number format, and bad concatenation. STL files can be viewed from any of the standard views (Figure 2) or the user can pan, rotate, and zoom via mouse commands.

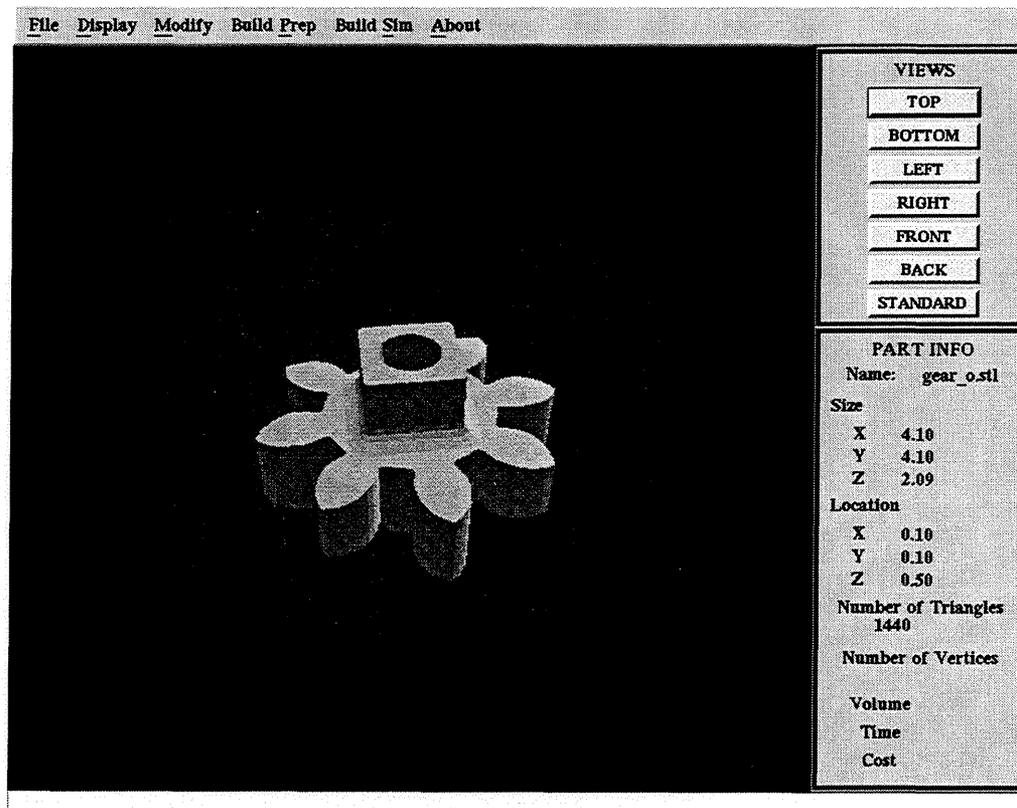


Figure 2. STL Viewing Screen.

Currently CIDES supports only flat shading of the image. Flat shading helps the user to detect errors in the STL file, especially when combined with what is called “disco lighting” by the users. This type of lighting is simulated by placing four different color light sources at the corners of a cube surrounding the part to give the part a multicolored appearance. Since the color

of a facet is dependent on its normal, any errors in the normal of a facet are quickly evidenced by a triangle of one color in the midst of a face that is a different color.

The second type of file read and written by CIDES is the SLI file, which is a 3D Systems Inc. slice or layer file. The dialog which allows the user to interact with these files is shown in Figure 3. SLI files are displayed layer-by-layer, allowing the user to turn on and off various vector types. For more rapid viewing, the user can choose to run up or down through the files, selecting the delay before each layer is replaced on the screen via the slider; delays range from no delay to half of a second. As with the STL file, the user can pan, zoom, and rotate with the mouse. The third type of viewable file is the V file, which are 3D Systems Inc. "merged" file used to build a part. These are also layer files, and are viewed using the same tools as the SLI files.

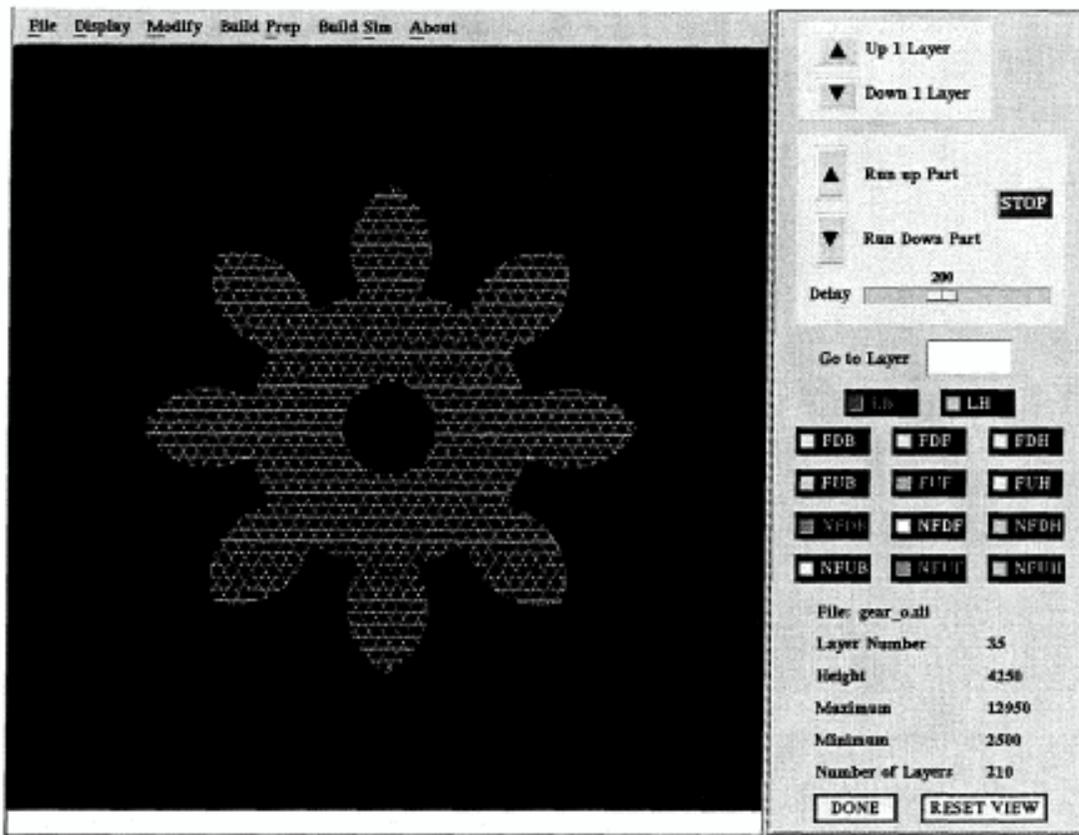


Figure 3. Layer Viewing.

CIDES also has several capabilities that help the user modify and manipulate files. Objects can be moved, rotated, and scaled in order to achieve the best build. The move feature includes both *Move*, which translates the part a relative distance along any of the coordinate directions, and a *Move To*, which sets the part minimum to a specified location. The latter is useful in order to assure that the part is in the positive octant. Another modification function sorts the facets of the STL file based on height for faster slicing.

The utility functions provide the user with capabilities to improve the SLA output or input new files. Verification routines check for and fix inverted normals and provide information on

missing facets. The support algorithm (Kirschman et al., 1991B) allows the user to customize a support structure and view it on the screen. The slicing routines are not as robust as their commercial variants, but the support can be sliced when generated, and a variety of hatch and fill capabilities are available. Volume information is provided for cost estimation. Algorithms are also included to translate from AutoCad neutral (DXF) files to STL files. CIDES offers sufficient capability to help the user of an RP system get more out of the system. The results of current research at Clemson University will be incorporated into future releases of CIDES to increase the utility of the software.

CONCLUSIONS

Rapid prototyping technologies are used to produce parts quickly directly from their CAD representations. However, many areas of the technology have not been made accessible to researchers wishing to “extend the envelope” of RP technology. CIDES includes a number of important features that have enabled Clemson University to increase the capabilities of the Rapid Prototyping (RP) systems that it has through software. CIDES provides input, output, and viewing capabilities for several common formats. It also provides facilities for STL file modification, support generation, slicing, and file translation. This has made CIDES a research testbed for a number of different projects at Clemson University. Currently CIDES is closely coupled with the SLA because that was the first available device, and the only one at Clemson University for several years. However, the code is being expanded to work with other RP systems, including the selective laser sintering testbed at Clemson University. CIDES 2.0 incorporates a new GUI which was created using sound GUI design principles.

ACKNOWLEDGMENTS

The authors thank NASA for its support of this research through NASA grant NAG3-2046. The authors also wish to acknowledge the contributions of Oliver Heim, Shannon Hughes, and Vishal Jadhav to the development of the CIDES 2.0 interface. Finally, the authors would like to thank all the users who were involved in the development and use of CIDES; their insights, comments and suggestions were the driving force of this project.

REFERENCES

- Frank, D., and Fadel, G. (1994). "Preferred Direction of Build for Rapid Prototyping Processes." *Proceedings of the Fifth International Conference on Rapid Prototyping* Dayton, OH.
- Gould, J. D. (1988). How to Design Usable Systems. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. (pp 757-789) Amsterdam: Elsevier Science Publishers.
- Gould, J. D., Boies, S. J., Levy, S., Richards, J. T., and Schoonard, J. (1987). “The 1984 Olympic Message System: A Test Of Behavioral Principles Of System Design.” *Communications of the ACM*, Volume 30, Number 9, September 1987, pp. 758-769.
- Kirschman, C. F., Bagchi, A., Jara-Almonte, C. C., Dooley, R. L., and Ogale, A. A. (1991A). “Stereolithographic Support Structure Design for Rapid Prototyping.” In *Proceedings of the Second International Conference on Rapid Prototyping* Dayton, OH.

- Kirschman, C. F., Jara-Almonte, C. C., Bagchi, A., Dooley, R. L., and Ogale, A. A. (1991B). "The Clemson Intelligent Design Environment for Stereolithography." In *Proceedings of Second the International Conference on Rapid Prototyping* Dayton, OH.
- Tata, K. (1995). "Efficient Slicing and and Realization of Tesselated Objects for Layered Manufacturing." Masters Thesis, Clemson University, Clemson, South Carolina.
- Terala, K. (1996). "Feed Forward Geometric Compensation for Selective Laser Sintering." Masters Thesis, Clemson University, Clemson, South Carolina.
- Ulrich, K. T. and Eppinger, S. D. (1995). *Product Design and Development*. New York: McGraw-Hill.
- Vuyyuru, P., Kirschman, C., Fadel, G., Bagchi, A., and Jara-Almonte, C. C. (1994). "A NURBS-Based Approach for Rapid Product Realization." *Proceedings of the Fifth International Conference on Rapid Prototyping* Dayton, OH.
- Wodziak, J, Fadel, G., and Kirschman, C. (1994). "A Genetic Algorithm For Optimizing Multiple Part Placement To Reduce Build Time." *Proceedings of the Fifth International Conference on Rapid Prototyping* Dayton, OH.
- The Clemson University Laboratory to Advance Industrial Prototyping (LAIP) can be reached through: Elaine Hunt, Director (elaine.hunt@ces.clemson.edu) by phone at (864) 656-4435 or via the World Wide Web at <http://design.ces.clemson.edu/rp/>.