A Methodology for Optimum Selection of Solid Modeling Software

Gül E. Okudan
The Pennsylvania State University

Abstract

This study proposes a methodology that would enable a design educator or a design practitioner to optimally select a solid modeling software (solid modeler) for varying objectives. Specifically, tasks accomplished to propose the methodology include: 1) reviewing past literature to compile the criteria used for selecting solid modelers, 2) preliminary comparison of a number of solid modelers on established criteria, 3) running designed experiments for comparing the user performance on predetermined solid modeling functions, and 4) compiling the experience gained as a generic methodology. The application was completed over a two-year period while a systematic selection process was undertaken at The Pennsylvania State University (Penn State). This paper documents the entire selection process including the student design performance data collected. The set of outcomes of the study is expected to aid companies and design educators in making solid modeler selection decisions.

Introduction

One of the necessities for a company to succeed in today's global competition is its ability to identify customer needs and to quickly create products that meet these needs. This necessity, which involves a set of activities beginning with the recognition of an opportunity and ending in the delivery of a product to the customer, is the rapid product development process. Rapid product development has been especially important since the late 1980s. There have been vast improvements in the area, mostly focused on searching for ways to shorten the development process duration. Among these, the advancement in design software is very significant. Accordingly, when preparing engineering students for similar responsibilities, integrating a solid modeler to design teaching is a must. However, it is not a trivial task. Associated with the integration, several questions need to be answered. For example, 1) Does the software have educational materials? 2) Are the educational materials adequate? 3) Is it easy and quick for students to learn? 4) Can the faculty gain the necessary knowledge and expertise to teach it in a short time? and 5) Does learning the software help students learn another solid modeling software easier? Beyond integrating a solid modeling software to design teaching, when one considers how a company might consider selecting a design software, more questions would arise such as: 1) Does the design software satisfy the needs of the product development process? 2) How efficient is it? and 3) How does it compare to similar products? These questions are important to answer when the goal is to shorten the product development process by utilizing cutting edge design software solutions.

In order to provide a decision tool for design educators and practitioners, this study proposes a methodology for optimally selecting a solid modeler for varying objectives. Furthermore, application results of the methodology are presented. The application was completed over a two-year period while a systematic selection process was undertaken at Penn State. This paper documents the entire selection process including literature review, the proposed solid modeler comparison methodology and its application results.

Literature Review

In order to 1) compile the criteria for use during solid modeler selection process, and 2) to review previous solid modeler comparison studies, a comprehensive literature search was completed. Below is a summary of the findings of the literature search.

Solid modeling was developed in the mid 1970s as a response to the need for informational completeness in geometric modeling, and can be defined as a consistent set of principles for computer modeling of three-dimensional solids (Shapiro, 2002). It uses the mathematical abstraction of real artifacts, which are transferred into computer representations (Requicha, 1980; Requicha and Voelcker, 1981). Solid modeling was intended to be a universal technology for developing engineering languages that must maintain integrity for 1) validity of the object, 2) unambiguous representation, and 3) supporting any and all geometric queries that may be asked of the corresponding physical object (Shapiro, 2002).

Early efforts in solid modeling focused on replacing manual drawings with the unambiguous computer models to automate a variety of engineering tasks (e.g., design and visualization of parts and assemblies, computation of mass, volume, surface area of parts, simulations of mechanisms, and numerically controlled machining processes (Requicha and Voelcker, 1982 and 1983; Voelcker and Requicha, 1993). Today, it is seen as an integral tool for product development because "... (it) allows everyone involved in the development of a new product—marketing/sales staff, shop-floor personnel, logistics and support staff, and customers—to add their input when changes can be made quickly and easily" (Schmitz, 2000).

Product design related applications of solid modeling are classified as 1) geometric design, 2) analysis and simulation, 3) dynamic analysis, and 4) planning and generation (Shapiro, 2002). Furthermore, while at one time the statement "... there are currently no CAD systems that live up to the requirements of the concept design" (Van Dijk Casper, 1995) was true, recent experimentation with solid modeling showed improvements in its usage for concept design (Tovey and Owen, 2000). It is also now commonly expected that solid modeling tools include collaborative tools allowing multi-location partners to work on the same design. Nam and Wright's (2001) recent paper includes a good review on design collaboration using solid modelers.

Overall, solid modeling impacts a great variety of concurrent engineering activities, and its importance is increasing due to its wide acceptance. The concurrent engineering activities that use solid modeling include design sketches, space allocation negotiations, detailed design, interactive visualization of assemblies, maintenance-process simulation studies, engineering changes, reusability of design components, analysis of tolerances (Requicha, 1993), 3D-mark-up and product data management, remote collaboration with internet catalogs of parts, electronic interaction with suppliers, analysis (e.g., mechanism analysis or finite elements), process planning and cutter-path generation for machining, assembly and inspection planning, product documentation and marketing (Rossignac and Requicha, 1999).

In a recent article, David Ullman (2001) discusses the current stage of computer aided design (CAD) systems as a design support system and indicates opportunities for software developers to bridge the gap between how designer activities can be supported better in the concurrent engineering realm. Some of these are: 1) an ability to visualize function before geometry is fully defined, 2) extending CAD systems to provide the designer with information about anticipated material and manufacturing methods, 3) generation of a running update of costs as parts and assemblies are changed in real time, and 4) integration of requirements and constraints into the development of parts and assemblies (Ullman, 2001). Despite these current inadequacies of solid modelers, a recent review of design software users survey (CAD Manager 2003 survey) showed that only 30.4% of the design practitioners are using 2D CAD systems. The rest are either using only 3D CAD (6.8%), implementing a hybrid usage of 2D/3D CAD systems (36.4%), or mainly using 2D CAD but evaluating 3D CAD (26.4%) (Green, 2003). This shows the trend in industry in adopting solid modeling software. Furthermore, due to its wide acceptance in industry, its integration to curriculum is changing the engineering/product design teaching (Barr et al., 2002).

Despite the apparent trend in adopting solid modelers in industrial and in educational institutions, selecting the solid modeler that is best suited to the task at hand is not an easy decision. One needs to consider several issues when making such a decision. In addition, one set of criteria that is suitable for one setting may not be for another. For example, criteria used to select a solid modeling software for a design company will differ when compared to the criteria used at an educational setting.

Previous work on solid modeling software comparison include 1) one CAD expert offering his review comments for various products without providing an established set of criteria, 2) rating a software using a predetermined set of criteria, and 3) comparing several similar software packages using predetermined criteria. For example, one can find solid modeler review and ratings in *Professional Engineer* and *CADENCE* (now *CADALYST*) magazines. To give examples: January 1993 issue of the *Professional Engineer* magazine includes a review on four different low cost CAD offerings by a CAD expert, where no particular review criterion is provided (Claypole, 1993). October 2003 issue of *CADENCE* contains a review of CATIA V5 R11. After its review, ratings are provided for the criteria including 1) installation and setup, 2) interface/ease of use, 3) features/functionality, 4) expandability/customization, 5) interoperability/web awareness, 6) support/help, 7) speed, 8) operating systems, and 9) innovation (Greco, 2003). In this sort of rating, there are several problems. For example, it is not possible to compare ratings of two different software completed by different experts. Because the way the experts have interpreted the criteria might be different. Even when the same person evaluates a number of different software, the potential bias the evaluator may have toward one application is very hard to eliminate. In fact, this problem was brought up by Martin and Martin (1994), and studied using published reviews and expertise of reviewers.

It is possible to eliminate the potential bias one can have towards one software by introducing expert users to the comparison. For example, Martin and Martin (1994), and Kurland (1996) invited various vendors to supply operators to partake in separate comparison studies. This way potential biases due to partiality towards one software over the other, or differences between software operators in terms of their skill levels were eliminated. However, in this case it is not clear if the solid modeler can be used by any user as effectively as the expert user partaking in the studies, after an adequate learning period. In other words, experimenting with an expert user cannot yield broader conclusions, because the graphical user interface (GUI) of the modeler can be interpreted differently by different users. Therefore, the GUI determines the overall usability of the modeler and the productivity of the user (Rossignac and Requicha, 1999).

In the 1980s, the introduction of icons and small pictures, which incorporated the desktop mouse as an input mechanism (Rheingold, 1991), changed the human-computer interaction (HCI). The implementation of this GUI takes advantage of the human capability to recognize and process graphical images quickly, and has become a universal HCI standard. Accordingly, most solid modelers use it today. However, the growth of interfaces is a concern for software developers because it might be a barrier in solid modeling education and in engineering practice (Jakimowics and Szewczyk, 2001). It is believed that the layout of GUI elements influences the way the user can interpret them (Ambler, 2000). While the user's correct mental model of the interface can help with his productivity, a false image of the interface might mislead them and limit their ability to work with the software effectively (Genther and Nielsen, 1996). For example, a recent experimental study showed that, if an unknown icon A in software 1 looked like a well-known icon B in software 2, the users supposed that the icon A represented the same function as the icon B, even if both pieces of software were quite different (Szewczyk, 2003). Therefore, it is clear that differences in user mental models of GUI is expected, and thus productivity differences may arise. This point makes it clear that any comparative study of solid modelers should involve multiple users being tested under similar circumstances. The methodology proposed in this paper overcomes the limitations in early comparison studies.

Proposed Methodology

The proposed methodology for solid modeler comparison, which is named as Solid Modeler Evaluation and Comparison Cycle, is given below in steps. Each step is then explained for its rationale for being a part of the methodology.

Solid Modeler Evaluation and Comparison Cycle (SMECC):

Step 1) *Develop a short list of solid modelers for comparison.*

This step is included to compile information to answer the question, "Which solid modelers are used by competitors, suppliers, and customers?" This information is important because in many cases design files need to be exchanged between suppliers and customers, and it is always important to know what the competitors are using. Attention should be given to the multiple solid modeler usage at various levels such as mid-level solid modelers versus high-level modelers. In general, price range of the solid modeler is a good indicator of its level. Therefore, various solid modelers should be clustered based on price ranges and compared within clusters.

Step 2) *Determine the solid modeling functions to be compared.*

Determining the solid modeling functions and criteria for comparison should be context specific because what is needed from the solid modeler depends on the specific applications of the unit that is looking into acquiring the modeler. Table 1 summarizes solid modeler comparison criteria and functions used for empirical testing or proposed for future testing by several practitioners and researchers. In the table, criteria empirically tested, and proposed are indicated by (T) and (P) respectively. Furthermore, for each of the criteria or function listed, a check mark is included to indicate in which previous studies it was proposed or tested. In addition, the number of solid modelers compared is shown in parentheses for each study indicated.

As seen in Table 1, due to the increasing importance of design collaboration because of globalization, outsourcing, and customization, a new set of proposed criteria is focused on collaboration effectiveness of solid modelers. However, published empirical comparison results were not found during the literature survey completed for this research. Therefore, a zero is placed in parenthesis to indicate that while the criteria have been proposed they were not used to compare solid modelers.

A subset of the criteria and functions (from Table 1) should be selected when comparing solid modelers.

Step 3) *Compile a training manual and a schedule for solid modeling learning for the functions determined in step 2.*

This compilation should be done using solid modeler's original training and support manuals, because it is assumed that the developers of the software are in the best position to provide training material. Training manual should cover the selected solid modeling functions and criteria for comparison in Step 2.

Step 4) *Conduct user performance experimentation.*

This step requires a number of users completing the training manual over a predetermined amount of time, and then assessing their performance. User performance should be measured for predetermined solid modeling functions (e.g., extrusion, sweep, revolve, assembly) and using various test problems. It should involve as many users as required to yield a reliable hypothesis testing. Variation in the collected data will be due to various aspects of human performance such as spatial and cognitive abilities, different interpretations of GUI elements by different users.

Step 5) *Analyze the user performance data statistically and conclude.*

Collected data should then be analyzed statistically to conclude with sufficient confidence.

Step 6) *Repeat steps 1-5 in regular intervals.*

The SMECC cycle should be repeated in predetermined intervals for continuously taking advantage of rapid developments in solid modelers.

SMECC Application at Penn State

As Rossignac (2003) acknowledged, there exists a gap between traditional research in any specific field, which is not concerned with educational objectives, and research in education, which is focused on fundamental teaching and learning principles. Accordingly he proposed Education-Driven Research (EDR) for simplifying the formulation of the underlying theoretical foundation and of specific tools and solutions to make them easy to understand and internalize. A similar point of view was taken at Penn State while developing a methodology to select a solid modeler that will enable effective learning without limiting the time to teach design knowledge. With this in mind, a comprehensive solid modeler comparison was initiated during Spring 2002, which was completed in two years. This section summarizes the steps of this two-year effort.

Step 1) *Develop a short list of solid modelers for comparison.*

For this purpose, using the list of top 30 engineering schools in year 2002 (provided by

Previous Studies of Solid Modeler Comparison (Author, year of publication, number of solid modelers compared)						
Comparison Criteria or Functions **Tested or Proposed (T or P)**	**Mackrell 1992, (0)**	**Martin & Martin 1994, (6)**	**Kurland 1996, (5)**	**Orr 2002, (0)**	**Greco 2003, (1)**	**Okudan 2004, (4)**
Extrusion (P, T)	√		√		√	√
Shelling/ skinning (P, T)	√		√		√	√
Creation of draft angles (P, T)	√		√		√	√
Filleting, chamfering/ blending (P. T)	√		√		√	√
Creation and retention of ribs (T)			√		√	
Feature patterns (linear, circular) (T)			√		√	√
Sweeping profiles along curves (P, T)	√				√	√
Lofting (P, T)	√				√	√
Revolve (P, T)	√				√	√
Associativity (one way, two way) (T)			√		√	√
Cross sections (T)			√		√	
Offset sections (T)			√		√	
Isometric views (T)			√		√	√
Assembling parts (T)			√		√	√
Parametric relationships altering (T)			√		√	√
Complex blends (T)			√		√	
Installation and setup (T)					√	
Ease of use (P, T)	√	√			√	
Speed (P, T)	√	√			√	√
Reliability (P, T)	√					
Cost (T)		√				√
Number of mouse operations to complete a predetermined object model (T)		√				
Operating system (T)					√	
Interface/command structure (T)		√				
Animation (T)		√				
Rendering (T)		√				
Dimensioning (T)		√				√
Innovation (T)					√	
Support/educational materials (T)					√	√
Customization (T)					√	
Web awareness (T)					√	
Document management (P)				√		
Viewing and markup (P)				√		
Threaded discussion (P)				√		
Requirements capture (P)				√		
Product data management systems (P)				√		
Calendar and task management (P)				√		
Whiteboard (P)				√		
Project directory (P)				√		
File conversion facilities (P)				√		
Polls (P)				√		
Decision making tools (P)				√		
Audit trail (P)				√		

Table 1 Solid Modeler Comparison Criteria and Functions

US News and World Report), a web search was completed to document the solid modeler (i.e. SolidWorks, Inventor, ProEngineer, etc.) usage. Search on each school's website included solid modeler usage in any of its engineering design courses, focusing primarily on mechanical engineering. To do this, first Mechanical Engineering Department's home page was targeted and then curriculum listings as well as any available course descriptions or syllabi were reviewed. Course descriptions proved to be of little help since they are somewhat broad and do not go into detail about the course. However, if a course syllabus was accessible, it usually listed what software was used for the solid modeling portion of the course. If neither a course description nor a syllabus was available, the school's website search engine was turned to as the next resource. Then, the website was searched for direct hits on keywords such as SolidWorks or ProEngineer. This resulted in a listing of any web page (on the school website) containing those keywords. From this list, a course web page containing the information needed could usually be found. As the last resort, individual course instructors were emailed.

After gathering, all data were compiled on a spreadsheet with each school's name in order of ranking in 2002, the engineering design course number and name, software used in the course, as well as the respective website from which the information was collected. Of the 21 schools from which data were available, 11 use ProEngineer, 10 use SolidWorks, 2 use Solid Edge, 2 use Inventor, 1 uses Alibre, 1 uses Mechanical Desktop, 1 uses CATIA, and 1 uses MATLab (Okudan, 2004). After reviewing these data, three solid modelers that were at comparable price levels and relatively widely used, were chosen for comparison: SolidWorks, Solid Edge and Inventor. While ProEngineer was in fact the most widely used solid modeler, it was not included in the study due to its cost. At the time, the solid modeling package that was being used at Penn State was IronCAD. With IronCAD, four packages were included in the comparison study: SolidWorks, Solid Edge, Inventor and IronCAD.

Step 2) *Determine solid modeling functions to be compared.*

Selected functions for comparison include extrude builds, extrude cuts, filleting, associativity of the solid modeler, dimensioning, isometric views and creating 2D drawings of solid models etc. For the complete list of selected functions see the last column in Table 1. These functions were compiled based on the requirements of the Introductory Engineering Design teaching at Penn State.

Step 3) *Compile a training manual and a schedule for solid modeling learning for the functions determined in step 2.*

The four solid modeler companies SolidWorks Inc., Autodesk Inc., IronCAD LLC, and EDS were contacted at the same time to provide training materials for the comparison study. All companies responded with a collection of their educational materials. Then, all training materials received were reviewed for their adequacy in supplementing the design teaching for the Introductory Engineering Design course at Penn State. This process eliminated two of the solid modelers originally selected to be in the short list for comparison[1], because their educational materials were not found to be adequate for implementation or integration to the course.

Step 4) *Conduct user performance experimentation.*

For the remaining two solid modelers, a classroom experimentation was planned to compare their effectiveness on students' solid modeling learning and hence modeling performance. The experimentation involved the same instructor teaching two sections of the same Introductory Engineering Design course, with one software in one section, with the other software in another section during the same semester.

The pre-prepared training manual for each modeler was designed to take about 20 hours in class-work for each student. These in class work hours were planned as 10 two-hour sessions over the semester. Sessions were conducted in a computer laboratory as a part of the six-hour Introductory Engineering Design course. On the seventh week of the training two CAD quizzes were given to both sections on the same day using the same questions. Students were given two hours to complete both quizzes. First quiz was given to all students at the same time in each section. As soon as a student was done with the first quiz, second quiz was given. None of the students had to wait for any other to start the second quiz.

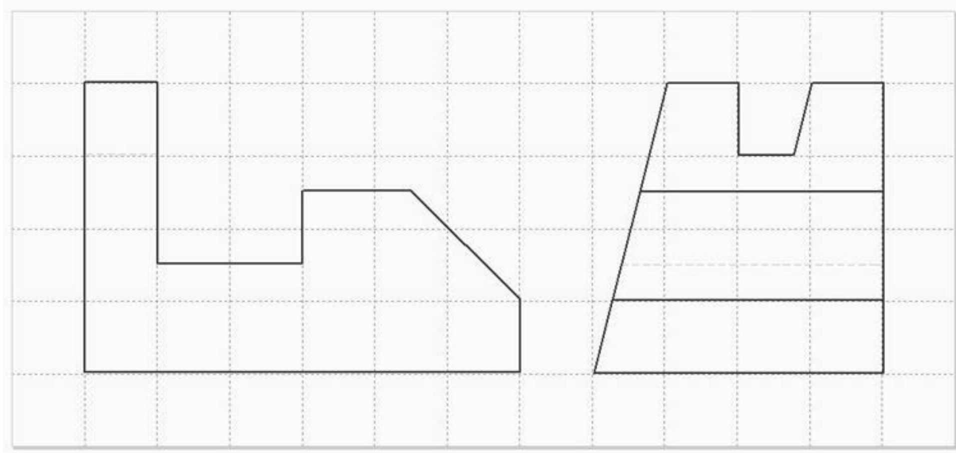

Figure 1 First quiz problem (Adopted from Bertoline et al.,1998, pp.353)

The CAD quizzes were given as practice quizzes three weeks before they completed their training manuals. Students were offered extra grades for participating in the study to motivate them for a high performance. Questions were designed to understand the student learning on the predetermined curriculum subjects[2], which include the software comparison functions. Two performance measures were used in this experimentation: 1) correctness and completeness of the solid modeling drawing (assessed by a performance grade between 0-1), and 2) time to complete the drawing in minutes. Figures 1 and 2 show the quiz questions respectively.

For both quizzes the following items were asked to be completed:

1. Create the 3D object using your solid modeler.
2. Create the standard multiviews and an isometric view on an A-size landscape paper.
3. Include scale information, your name and drawing name in the title block.
4. Complete dimensioning and print your work.

During the quizzes, students were not allowed to ask questions or talk to each other. Furthermore, they were asked to run only the solid modeler on their computer. Table 2 shows the results of this experimentation. The quizzes were taken by all students on identical computers in the same computer laboratory. The first section of students completed their quizzes between 12:20-2:20 pm, and the next section at 2:30 pm-4:30 pm. Students were not allowed to take the quiz questions with them when they were done.

Step 5. *Analyze User Performance Data Statistically And Conclude.*

Using Minitab™ Release 13.1, differences of sample averages for user performance and completion time for both quizzes were tested for their significance. Table 3 shows these data. As can be seen with the p values for all four two-sample t tests, differences in sample means were not found to be statistically significant. This means that for the functions that were the subject of comparison, both software deliver similar results with a similar average time for students to complete the same problems.

In fact, when data in Table 2 were analyzed, it is seen that with the exception of a few cases, users were able to complete both quiz problems correctly. This is reflected in the performance data for both samples mostly being 1.00 out of 1.00. However, while the sample means were not statistically significant, the time it took users to complete the quizzes had a spread for both solid modelers. Therefore, using Minitab™ Release 13.1, variance tests were conducted for completion time of quizzes.

When the hypothesis test for equality of variances between two samples for quiz 1 using an F-test is completed, sample variances were found to be significantly different. The test is conducted for 95% confidence level. Figure 3 shows the Minitab output with a p value of 0.031.

The significant difference in sample variances indicates a more homogeneous user performance data (similar in completion times) (in this case for software 2), in comparison to a more heterogeneous set (software 1). This might be a sign

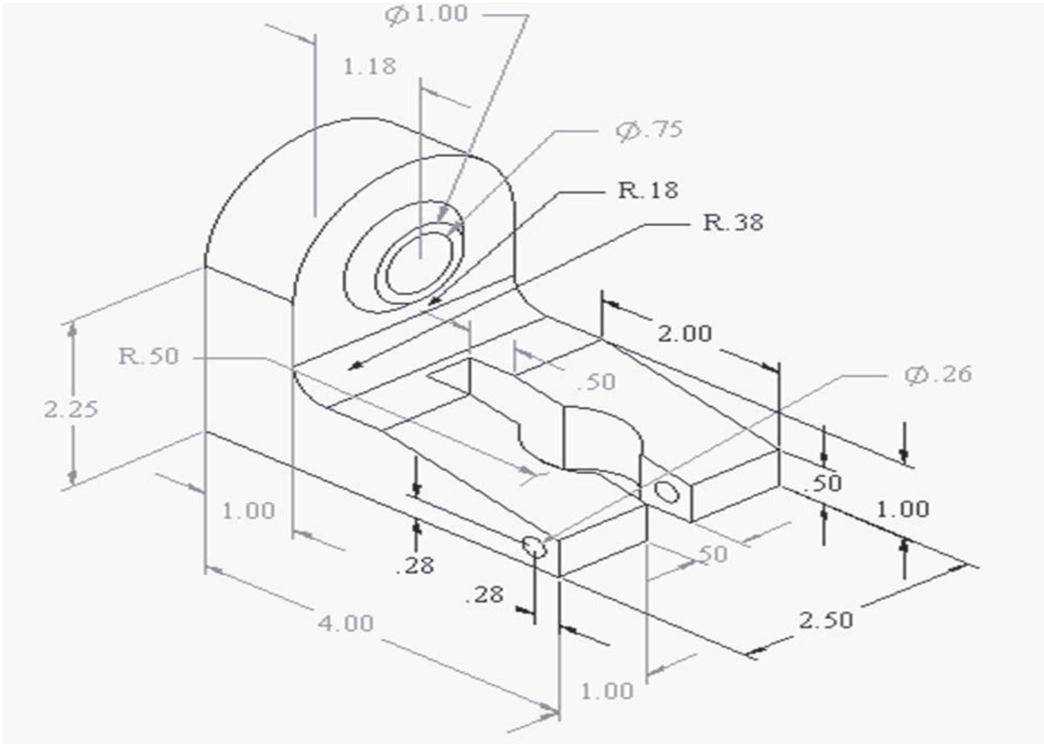

Figure 1 Second quiz problem (Adopted from Bertoline et al., 1998, pp.362)

Performance for Quiz1, Software1 PerQ1S1 (Grade 0-1)	Time Spent for Quiz1, Software 1 TimeQ1S1 (Min)	Performance for Quiz2, Software1 PerQ2S1 (Grade 0-1)	Time Spent for Quiz2, Software 1 TimeQ2S1 (Min)	Performance for Quiz1, Software2 PerQ1S2 (Grade 0-1	Time Spent for Quiz1, Software 2 TimeQ1S2 (Min)	Performance for Quiz2, Software2 PerQ2S2 (Grade 0-1)	Time Spent for Quiz2, Software 1 TimeQ2S2 (Min)
1.00	30	1.00	65	1.00	30	1.00	35
1.00	25	0.75	60	1.00	15	0.75	40
1.00	60	0.80	30	0.90	20	0.90	90
1.00	30	1.00	60	1.00	30	1.00	50
1.00	60	1.00	60	1.00	25	1.00	55
1.00	30	1.00	25	0.75	30	1.00	30
0.75	45	1.00	50	1.00	35	1.00	45
1.00	24	1.00	54	1.00	20	1.00	20
0.50	30	1.00	20	1.00	20	0.75	55
1.00	15	1.00	30	1.00	30	1.00	40
1.00	15	1.00	25	1.00	30	1.00	25
1.00	15	1.00	30	1.00	10	1.00	20
1.00	15	1.00	45	1.00	23	1.00	33
1.00	35	1.00	30	1.00	30	1.00	40
1.00	12	1.00	45	1.00	30	1.00	40
1.00	25	1.00	48	1.00	40	1.00	25
1.00	50	1.00	55	1.00	20	1.00	76
1.00	50	1.00	70	1.00	20	1.00	30
1.00	20	1.00	60	1.00	45	1.00	45
1.00	40	1.00	60	1.00	20	1.00	50
1.00	15	1.00	45	1.00	45	1.00	65
1.00	15	1.00	30	1.00	41	1.00	56
1.00	09	1.00	45	0.75	15	1.00	25

Table 2 User Performance Experimentation

T-Test	N	Mean	Standard Deviation	Result
Two sample T test for PerQ1S1 vs. PerQ1S2	23	0.967	0.114	Estimate for difference: -0.0065 95% CI for difference: (-0.0637, 0.0507) T-Test of difference = 0 (vs. not =): T-Value = -0.23 P-Value = 0.819 DF = 44 Both use Pooled StDev = 0.0962
	23	0.9739	0.0737	
Two sample T test for TimeQ1S1 vs. TimeQ1S2	23	28.9	15.4	Estimate for difference: 1.78 95% CI for difference: (-5.85, 9.41) T-Test of difference = 0 (vs. not =): T-Value = 0.47 P-Value = 0.640 DF = 44 Both use Pooled StDev = 12.8
	23	27.13	9.61	
Two sample T test for PerQ2S1 vs. PerQ2S2	23	0.9804	0.0653	Estimate for difference: 0.0065 95% CI for difference: (-0.0348, 0.0479) T-Test of difference = 0 (vs. not =): T-Value = 0.32 P-Value = 0.752 DF = 44 Both use Pooled StDev = 0.0696
	23	0.9739	0.0737	
Two sample T test for TimeQ2S1 vs. TimeQ2S2	23	45.3	14.9	Estimate for difference: 2.26 95% CI for difference: (-7.48, 12.00) T-Test of difference = 0 (vs. not =): T-Value = 0.47 P-Value = 0.642 DF = 44 Both use Pooled StDev = 16.4
	23	43.0	17.7	

Table 3 Statistical Analysis of Results

of users' different interpretations of GUI elements and hence related performance differences. For example, in this experimentation, higher variance in completion time data for software 1 could be seen as a result of a higher potential for software 1 GUI elements' less than uniform interpretations. However, normality of the two sample data should be investigated first before pointing at solid modeler differences. Figure 4 and 5 show Anderson-Darling normality tests for both samples.

As can be seen in Figures 4 and 5, for a 90% confidence interval both of the data sets follow normal distributions (p-value for TimeQ1S1= 0.029, and p-value for TimeQ1S2= 0.087). Thus the significant difference in variances cannot be dismissed.

When a hypothesis test for equality of variances between two samples using an F-test is completed for the second quiz problem, sample variances were not found to be significantly different. The test was conducted for 95% confidence level using Minitab. Figure 6 shows the Minitab output with a p value of 0.430.

Based on the statistical analysis presented above, in terms of time to complete the problem while no significant difference in sample means for both quiz problems, and no significant difference in variances for quiz 2 were found, due to the significant variance in time to complete the problem for quiz 1, software 2 is selected to be integrated to the engineering design curriculum. Because based on the multi-user experimental study presented above, software 2 was deemed to yield a more uniform user performance in terms of completion time when compared to software 1.

Step 6) *Repeat steps 1-5 in regular intervals.*

Because the application took approximately two years, the interval for repeating the SMECC cycle is determined to be two years. The new cycle has started in Spring 2004 semester.

Conclusion

Solid Modeler Evaluation and Comparison Cycle (SMECC), as a methodology that eliminates the limitations of previous solid modeler comparison studies is proposed and its application is shown in this paper. Steps of the cycle are:

Step 1) Develop a short list of solid modelers for comparison. In this study cost was used as a criterion to select a subset of solid modelers from

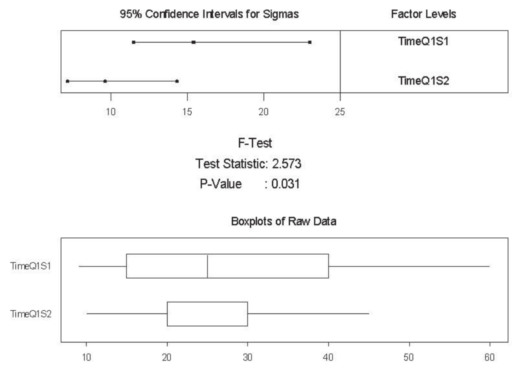

Figure 3 Variance Test Results for Quiz 1 Completion Time Data

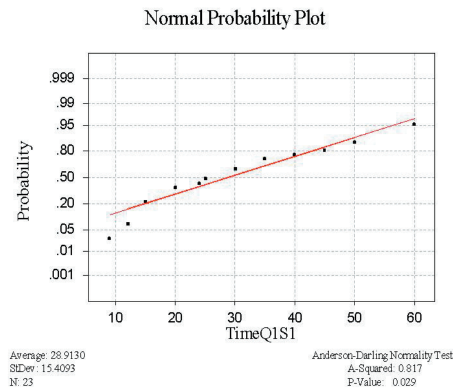

Figure 4 Normality Test for TimeQ1S1 Data

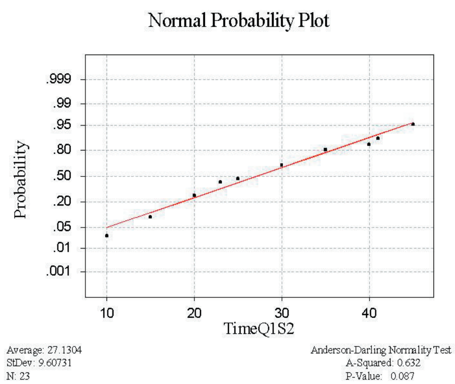

Figure 5 Normality Test for TimeQ1S2 Data

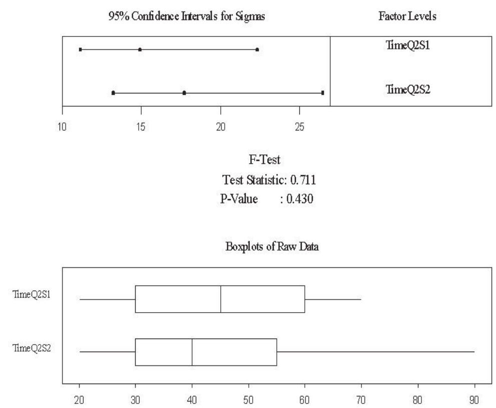

Figure 6 Variance Test Results for Quiz 2 Completion Time Data

the compiled list of top 30 engineering schools' choices, however, other criteria could be used for different situations. For example, existing computer hardware and/or platform might bring limitations, or availability of specific functions such as animation, might qualify or disqualify solid modelers from the short list.

Step 2) Determine the solid modeling functions to be compared. Function determination can only be done with a clear idea of what the solid modeler is going to be used for. For example, during the Introduction to Engineering Design course, for which the selection study was completed, multi-location design collaboration is not required, therefore, viewing and markup capabilities of modelers were not compared. Table 1 provides a comprehensive list of solid modeling functions to choose from based on potential needs.

Step 3) Compile a training manual and a schedule for solid modeling learning for the functions determined in step 2. A clear, concise, mistake-free training manual with adequate number of examples, and exercises is very important in solid modeling learning. Most solid modeler companies have educational branches that prepare and maintain these training manuals in print or on-line medium. However, based on the selection experience presented in this article it can be stated that there is a significant difference among manuals of different software for the same functions in terms of clarity, conciseness and the number of mistakes; therefore, educational materials should be carefully reviewed.

Step 4) Conduct user performance experimentation. While there are other ways of selecting software such as using one expert evaluating a number of software packages, different experts evaluating different software using the same set of criteria etc., for institutions where the modeler will be used by a large number of people with different backgrounds (like educational institutions), a multi-user performance experimentation should be completed. This experimentation reveals potential performance differences due to the different graphical user interface items (menus, icons, etc.), and hence a more equitable performance field can be established for users. If, however, there are published user performance studies available for the same set of modelers for the same set of functions, experimentation is not needed.

Step 5) Analyze the user performance data statistically and conclude.

Step 6) Repeat steps 1-5 in regular intervals.

This solid modeler evaluation and comparison cycle (SMECC) has been developed based on a comprehensive review of the previous solid modeler comparison studies, comparison criteria and functions. In addition, it has been applied at Penn State and the experience gained is explained above. Overall, SMECC overcomes limitations of previous solid modeler selection studies in the literature.

Because solid modeling is becoming the choice of designers instead of 2D CAD software, the set of outcomes of this study such as the compiled set of solid modeler comparison criteria, the methodology proposed and applied for solid modeling software comparison are expected to aid companies and design educators in making better design software selection decisions.

References

Ambler, S.W. (2000). *User Interface Design: Tips and Techniques.* [http://www.ambysoft.com/userInterfaceDesign.pdf], retrieved December 2003.

Barr, R.E., Krueger, T. J. and Aanstoos, T.A. (2002). The New Digital Engineering Design and Graphics Process. *Engineering Design Graphics Journal, 66* (3), 6-11.

Bertoline, G., Wiebe, E.N. and Miller, C.L. (1998). *Fundamentals of Graphics Communication,* 2nd Edition, The Mc-Graw Hill Companies, Inc.

Claypole, J. (1993). Low Cost CAD - Reviewing the Latest Offerings. *Professional Engineer,* January, 12-13.

Gentner, D. and Nielsen, J. (1996). The Anti-Mac Interface. *Communications of the ACM, 39* (8), 70 - 82.

Greco, J. (2003). Feature Review: Catia V5 R11. *CADENCE*, October.

Green, R. (2003). CAD Manager Survey - Part 2. *CADALYST, 20* (12), 26-27.

Jakimowicz, A. and Szewczyk, J. (2001). Multi-user Interface Problems in Current CAD Systems. In *ACCOLADE Architecture Collaboration Design*, edited by M. Stellingwerff and J. Verbeke Delft, The Netherlands: Delft University Press Science, 189-200.

Kurland, R. H. (1996). *Comparing Mid-range MCAD Modelers*. [http://www.tehnicom.com], retrieved December 2003, 1-5.

Mackrell, J. (1992). Solids-Based M-CAD Systems-Selection and Testing. *AUTOFACT'92 Conference Proceedings,* September 10-12, 6.1-21, Detroit, MI.

Martin, R. and Martin, C. (1994). Comparing Mechanical CAD Programs, *Computer-Aided Engineering*, June, 44-50.

Nam, T.J. and Wright, D. (2001). The Development and Evaluation of Syco3D: A Real-Time Collaborative 3D CAD System, *Design Studies, 22*, 557-582.

Okudan, G.E. (2004). Selecting Solid Modeling Software for Integration to Engineering Design Teaching: Methodology & Application Results, *Proceedings of the 2004 American Society for Engineering Education Conference & Exposition.*

Orr, J. (2002). Mechanical Design: Take Your Team Online, *CADALYST, 19*(2), 20-30.

Requicha, A.A.G. (1980). Mathematical Models of Rigid Solids: Theory, Methods and Systems, *ACM Computing Surveys, 12*(4), 437-464.

Requicha, A.A.G. and Voelcker, H.B. (1981). An Introduction to Geometric Modeling and Its Applications in Mechanical Design and Production, In *Advances in Information Systems Science*, Ed. J.T.Tou, 8, Plenum Publishing.

Requicha, A.A.G. and Voelcker, H.B. (1982). Solid Modeling: A Historical Summary and Contemporary Assessment, *IEEE Computer Graphics and Applications, 2* (6), 9-24.

Requicha, A.A.G. and Voelcker, H.B. (1983). Solid Modeling: Current Status and Research Directions, *IEEE Computer Graphics and Applications, 3*(7), 25-30 and 32-37.

Requicha, A.A.G. (1993). Solid modeling and its applications, *Proceedings of the 19th NSF Design & Manufacturing Systems Conf.*, Charlotte, NC, January 6-8, 1993.

Rheingold, H. (1991). *Virtual Reality,* New York: Simon and Schuster.

Rossignac, J. (2003). Education-Driven Research in CAD, *GVU Technical Report GIT-GVU-03-29*, ftp://ftp.cc.gatech.edu/pub/gvu/tr/2003/03-26.pdf, retrieved May 2004.

Rossignac, J. and Requicha, A.A.G. (1999). Solid Modeling, In the *Encyclopedia of Electrical and Electronics Engineering*, Ed. J. Webster, John Wiley and Sons.

Schmitz, B. (2000). Tools for Innovation, *Industry Week*, May 15th.

Shapiro, V. (2002). Solid Modeling, In *Handbook of Computer Aided Geometric Design*, Eds. G. Farin, J. Hosheck, M.S. Kim, Elsevier Science Publishers.

Szewczyk, J. (2003). Difficulties with the Novices' Comprehension of the Computer-Aided Design (CAD) Interface: Understanding Visual Representations of CAD tools, *Journal of Engineering Design*, *14*(2), 169-185.

Tovey, M. and Owen, J. (2000). Sketching and Direct CAD Modeling in Automotive Design, *Design Studies, 21*, 569-588.

Ullman, D.G. (2001). Toward the Ideal Mechanical Engineering Design Support System, *Research in Engineering Design, 13*(2), 55-64.

Van Dijk Casper, G.C. (1995). New Insights in Computer-Aided Concept Design, *Design Studies*, *16*(1), 62-80.

Voelcker, H.B. and Requicha, A.A.G. (1993). Research in solid modeling at the University of Rochester: 1972-1987, in *Fundamental Developments of Computer-Aided Geometric Modeling*, Ed. L. Piegl, Academic Press Ltd., London, England, 203-254.

[1]The two solid modelers eliminated from the comparison study are not named, because the comparison methodology presented in this study is provided as a decision making tool for an objective and optimum solid modeler selection for specific uses. It is realized that different solid modelers will meet various customer needs at different levels. Therefore, software names were given in order not to mislead readers.

[2] CAD quizzes were not designed to measure their learning on revolves, sweeps and patterns functions. Because at the time of the practice quizzes, students were not done with those chapters. Furthermore, while they were done with the assemblies section of their manual, due to time limitations quiz questions did not include an assembly.