

Continuing Evolution of a Web-based Engineering Graphics Tutorial: Interactive Input and Response

P. E. Connolly and K. R. Maicher
Purdue University

Abstract

This paper explains the ongoing evolution and advances made with a computer-based tutorial, including the incorporation of an interactive input and response capability that allows the user to develop multiview drawings and receive detailed, real-time feedback as part of the tutorial function. A review of the original purpose and pedagogy of the tutorial project are included, as well as future plans for the tutorial and related efforts. Formal experimental plans, procedures, and results validating the tutorial's effectiveness will be presented and discussed.

Introduction

The challenge for many introductory computer graphics students remains the lack of ability to visualize in three-dimensional space. This crucial talent is one that professionals in the engineering and graphics realms must master in order to successfully accomplish the design projects and problem-solving tasks that they will face throughout their careers.

One practical application of visualization skill comes in the form of multiview drawing and development. Computer Graphics Technology and Engineering majors at Purdue University are required to learn and master this skill. Through the proper application of multiview drawing principles, students can improve their spatial acuity and comprehension by becoming more familiar with manipulating objects in 3-dimensional space.

Previous papers by the authors (Connolly & Maicher, 2003; Maicher & Connolly, 2003) have described the rationale for, and development of, an interactive Web-based tutorial for assisting students in the mastering of the principles of multiview projection and documentation, as well as enhancing visualization ability. The initial tutorial was developed using basic multimedia capabilities and options, and was revised using more advanced media authoring techniques and capabilities into a significantly more user-friendly and interactive product.

Tutorial Evolution - A Review Multiview Drawing Theory

As has been stated previously by the authors, the purpose of multiview projection is to produce

a series of two-dimensional images of a given three-dimensional object. Each 2-dimensional view represents a unique orientation of the target object. The development of an adequate number of these 2-dimensional views makes it possible to fully describe the object, allowing it to be accurately produced and used in various applications (Bertoline & Wiebe, 2003). The understanding and proper application of accepted rules and standards governing the creation of multiview drawings are an important segment of the tutorial instruction. A fundamental understanding of these guidelines is necessary for the student to utilize multiview drawings as a tool to enhance visualization and comprehension of three-dimensional objects.

Computer-based Instruction

The use of a computer-based tutorial to present the content material is logical based on its ability to deliver complex material, a high level of user interactivity, user control, adaptability, and individualized instruction pacing (Allesi & Trollip, 2001; Eom & Reiser, 2000; Newby, Stepich, Lehman, & Russell, 2000). Although computer-based instruction has somewhat mixed support in educational circles, it has been shown as an effective tool in many situations if appropriately designed and implemented (Holliday-Darr, K., Blasko, D. G., & Dwyer, C., 2000; Poli, Fisher, Pollatsek, & Woolf, 2003).

Web Development Tools

The first tutorial utilized simple HTML code and Javascripting, and was created with Macromedia Dreamweaver. The second version

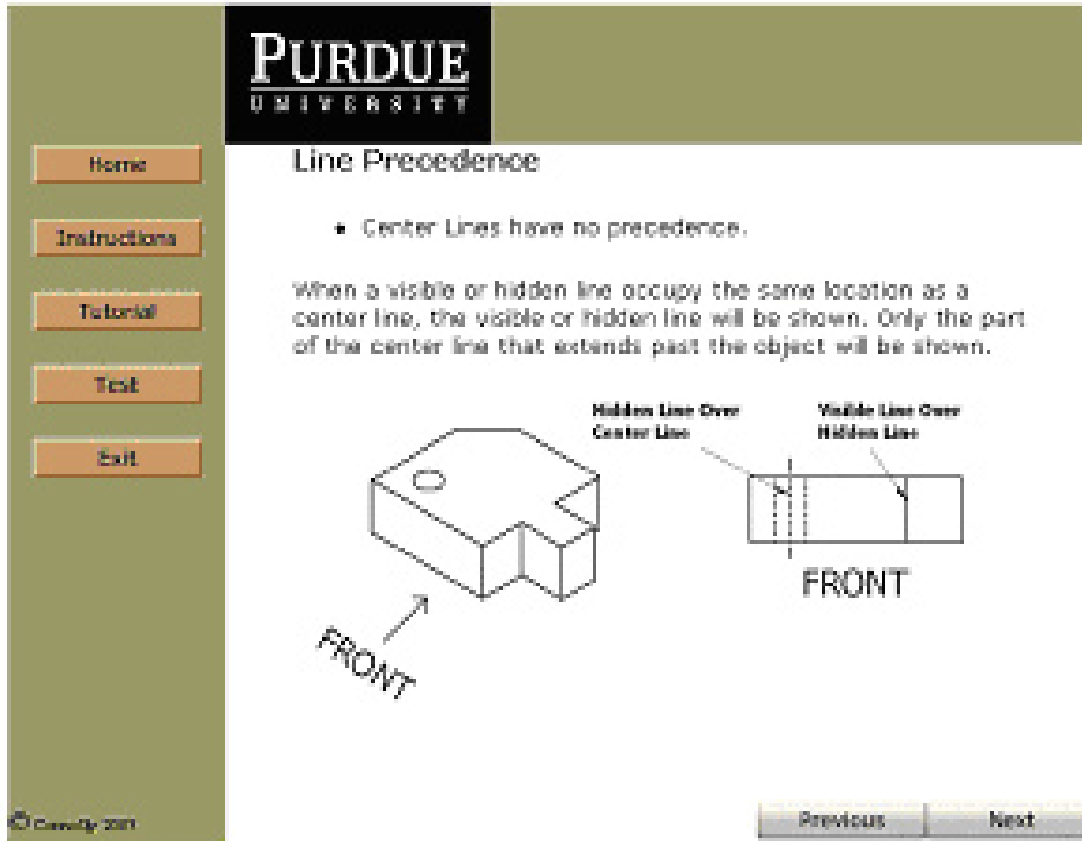


Figure 1 Initial Tutorial

of the application was developed with Macromedia Flash MX, Microsoft Access, and Macromedia Cold Fusion MX in order to integrate dynamic graphical elements and an integrated database from which to draw tutorial questions, answers, and graphical components. The current version of the application uses XML along with more sophisticated Actionscript and Cold Fusion programming elements.

Macromedia Flash MX 2004 was used because 1) it provided a great deal of flexibility in creating the interface elements, 2) had a robust native scripting language, and 3) contained a Drawing API that allowed us to create the Web-based drawing elements. Actionscript 2.0 was used to develop both the drawing and grading component of the application, while Cold Fusion MX 6.1 was used to communicate with the Microsoft Access database. XML documents containing user and geometry data were generated by Cold Fusion and used as an intermediate database. Using separate XML documents containing smaller portions of the database allowed the Flash-based program to function more efficiently.

Tutorial Description

The initial tutorial program consisted of three main sections: introduction, learning module (tutorial), and mastery test. An example of one of these tutorial pages is shown in Figure 1. These sections provided instructions on using the program, learning content, drill and practice reviews, and assessment on content mastery. The tutorial was evaluated via various means including content expert review, alpha test review for usability, a beta test for content appropriateness, and aesthetic input from web designers. Results from these assessments showed that the tutorial appeared to be effective as an instructional instrument, but was lacking in areas of structure, aesthetics, and usability.

The evaluation of the original multiview tutorial led to the decision to redesign the tutorial using a different authoring tool, Macromedia Flash MX. Although the general organization of the tutorial content was kept intact, the content information was condensed to make more effective use of screen space. While the simplicity of the navigation was retained, an effort was made

to provide greater flexibility and control by giving users more freedom to access multiple points in the site from various locations. Furthermore, the color scheme and layout of the tutorial were revised through the careful application of brighter colors and better graphics. The revised tutorial also included limited three-dimensional viewing capability, allowing students to manipulate models for improved visualization. One critical limitation that was identified with the early versions of the tutorial was the lack of ability of the tutorial to allow for any sort of electronic ‘drawing’ of the 2-dimensional multiview images. All drill and evaluation tasks in the tutorial consisted of multiple choice options for selecting the correct 2-dimensional representation. It was decided that the ability for students to electronically create multiview images would be the main focus for the next iteration of the tutorial, and would require significant resources and programming expertise.

Interactive Input and Response Capability

Overview

The interactive input and response tool was developed to provide a more realistic and effective learning experience for the students. It is designed to be an integral part of the tutorial, as a significant portion of the drill and practice and mastery assessment sections. Visually, the tool consists of two distinct areas, as shown in Figures 2 and 3.

The left portion is a pictorial representation of a simple part on an isometric grid; the right is a blank square grid for the students to use in

creating the requested view. Across the top of the screen are selection buttons, instructions, and icons that the students utilize to select problems and complete the assignments, as shown in Figures 4 and 5.

The selection button titled, “worksheet” contains libraries of pictorial images that can be selected by number by the students. As the pictorial image is selected, the program provides instructions to create the top, front, or side view. This requested view is a random generation that the user does not have control over. There are a series of icons that are to be used in the drawing creation. Currently, only the icons representing line creation, erase, clear all, show answer, and grade drawing functions are operational. The toggle icon for hidden line font is also functional. The user selects the desired icon with the left mouse button and uses the mouse to draw lines and select entities for erasing. The “Grade Drawing” icon checks the user-created 2-dimensional view and provides feedback to the error on the location and nature of potential errors. This is done through the use of color-coded text and color-highlighted lines. The “Get Answer” icon displays the solution in a green color. Appendix A contains several examples of the drawing tool and possible interactive responses.

Interactive Tool Development

The first step in developing the application was designing the drawing and grading elements, which were created in order to satisfy the most immediate needs discovered in testing the previ-

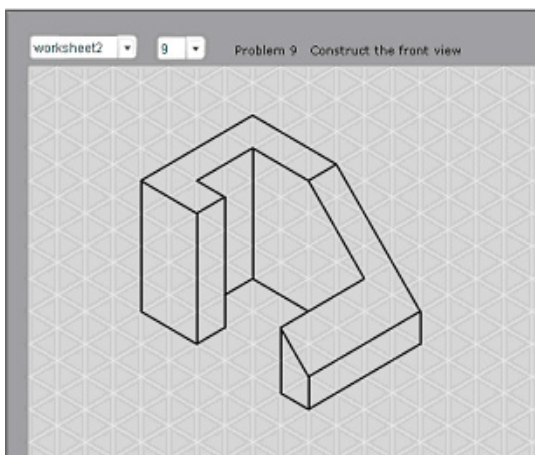


Figure 2 Left portion of tutorial

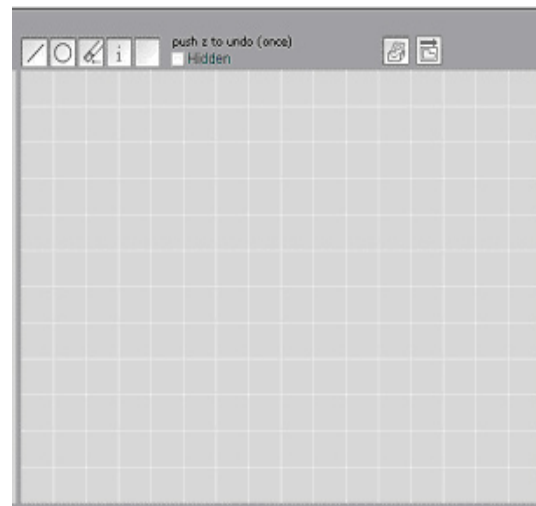


Figure 3 Right portion of tutorial

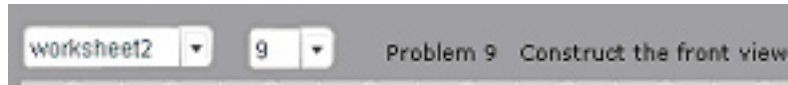


Figure 5 Left interface tool set

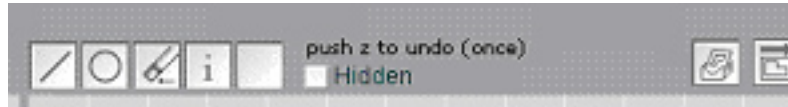


Figure 4 Right interface tool set

ous version of the application. Once the basic functions of the drawing and grading components were determined, the graphical interface elements were created and programming commenced.

The Drawing Program

The program utilized Flash drawing methods inherent to Actionscript to create the drawing tools such as lines, hidden lines, circles, and eraser; as well as the grid that tools “snap to” on the internal coordinate system. As the drawing tools are used, the program stores various line element properties such as starting coordinates, ending coordinates, and line type. These coordinates are temporarily stored in Actionscript arrays, which are accessed when sending the coordinates to the database for storage or when used by the grader for performance feedback.

In the administrative section of the program, the problems and answers drawn by the instructor/administrator are stored in Actionscript arrays then sent to the MySQL database via Cold Fusion. In the user section, students’ answers are again stored in Actionscript arrays, but instead of being sent to the server they are accessed by the grader to compare “student answer coordinates” to “instructor answer coordinates”.

The eraser feature also makes use of the stored coordinates by comparing the clicked point with stored lines in the Actionscript arrays. If the user clicks on a point that has been stored as part of a line segment, the program simply removes the corresponding line from both the screen and the array.

The Grader

While also utilizing array coordinate comparisons, the function of the grader is more complex than the drawing program and utilizes a series of three primary comparative coordinate tests to validate student answers. There are also several smaller tests implemented throughout the grading

process, which combined together give an accurate comparison between students’ solutions and the correct answers stored in the database.

In the first test the program analyses the slope, length, and visibility of each of the lines the user has drawn. Any line coordinates drawn by the user that do not correspond to stored answer coordinates are assigned an “incorrect” value. An error message is then prepared and stored, which will be displayed to the user on completion of the grading process. User line coordinates that match database coordinates are assigned a “correct” value and are used in the next test.

The second test compares the slope and lengths of the joined user lines relative to one another, and then compares those associations to the corresponding answers in the database. Since this method does not rely on the position in the coordinate system as a comparative reference, users and administrators are free to begin drawing their objects anywhere on the grid rather than a pre-defined starting point. If the correct slope/length associations are not met, the corresponding lines are marked as “incorrect” and an error message is prepared for the user. As with the first test, lines with corresponding coordinates are assigned a “correct” value and are used in the final test.

The final test consists of locating drawn elements that are unconnected to other elements (indicating a broken line). These elements are identified based on their position relative to lines that have already been deemed “correct” by previous tests. Lines who fail to meet the necessary criteria are again assigned an “incorrect” value, and an error message is prepared and stored. Upon completion of the grading, the stored error messages are retrieved and displayed to the user as either “incorrect”, “missing”, or “disconnected”. The program also retrieves the incorrect line values and highlights them in orange (incorrect) or red (disconnected).

Feedback

The interactive drawing and response tool was presented to approximately eighty students in an introductory engineering graphics course at Purdue University. These students had already been exposed to the basic principles of multiview drawing and had completed several simple problems translating 3-dimensional images into 2-dimensional multiview layouts. They were not given the entire tutorial of instructional content, but were asked to complete ten problems in the interactive module and were given an optional assignment to complete a seven-question survey regarding the interactive tool. (See Appendix B for a listing of survey questions). Forty students returned the anonymous questionnaire. Although in-depth analysis of the qualitative responses is not yet complete, there were several trends that were noted in cursory review.

- The students overwhelmingly found the tool to be easy to use and understand.
- They thought the error messages were too vague and, in some cases, inaccurate or misleading.
- They found the product to be helpful in understanding multiview drawing, and enjoyed using it.
- They did not like the color scheme (white on gray), or limitations on how lines could be drawn.
- The instant feedback (grading option) was greatly appreciated.

Some of the positive comments that were received are shown:

“Easy to get started, easy to use, helps build a foundation for fundamentals.”

“Anything that gets someone to think about things spatially and lets them do it themselves helps.”

“I was actually able to interact and change my models and it is a lot simpler and less time consuming than pencil and paper.”

“Very easy to use and would be extremely helpful to first-timers!”

“A clear and spiffy interface, easy-to-use

tools, and instant responsiveness.”

“It was really cool, and saves us a lot of time and \$6 iso-ortho paper. . . . Add music!”

“It was actually kind of fun. I would be willing to do this in my free time at home (if I had any).”

Although the feedback is superficial at this point, the authors are encouraged by the positive nature of the comments. The students also provided good suggestions for areas of improvement in content, functionality, and aesthetics. It was especially rewarding to see comments regarding ease-of-use, interactivity, and improved comprehension as a result of the tool usage.

Future Plans

Future plans for the tutorial include corrections and improvements to the interactive drawing tool to make it more effective and to expand its application. Some of these planned improvements are as follows:

- Implement a scoring mechanism to award points for answers and to provide a final score
- Implement a system for registration and storage of user data including personal info and scores. This will allow students to track improvements in their performance (this version)
- Implement additional, more sophisticated drawing tools
- Enhance the program so that it can recognize and grade more complex images with higher tolerances.
- Perform rigorous usability and performance testing on current application (this version)
- Convert the existing code to a more condensed, efficient OOP structure to facilitate modularity and implementation in future applications
- The ability to import images created in 3D programs such as AutoCAD for implementation into the image library.

More extensive qualitative and quantitative testing will take place with the goal of validating the capabilities of the tutorial as an educational application. As this occurs, it is expected that the

tutorial will be made available to other institutions in order to expand the quantity of the tutorial problems that the interactive tool has available. There are also tentative plans to create additional tutorials of similar nature for related topics in the engineering graphics realm.

Conclusion

At first glance, the benefits of this application may not be immediately apparent, given the vast numbers of interactive educational products currently on the market. For one to truly appreciate the potential advantages of this application, a comparison to the traditional method of multiview education is needed.

With traditional multiview education, educators are required to either produce new representations or draw upon existing images. The images must then be rendered in a viewable medium for students to see, and this medium must be easily distributed. The students must then attempt to create the correct views on paper or in some drawing program. The papers must then be collected or the digital files stored in a central location and painstakingly reviewed one-by-one by the educator.

With the multiview application, 3D objects and their corresponding orthographic views only need to be rendered once. From then on, they are stored in the applications database for future retrieval. Since the program is Web-distributed and uses one central location for data storage, multiple administrators can conceivably build an extensive library of images over time that can be re-used indefinitely.

The problem of constantly recalling and distributing the images on paper or some other medium is eliminated, since student will only need to access the application from any Web-accessible computer regardless of geographic location.

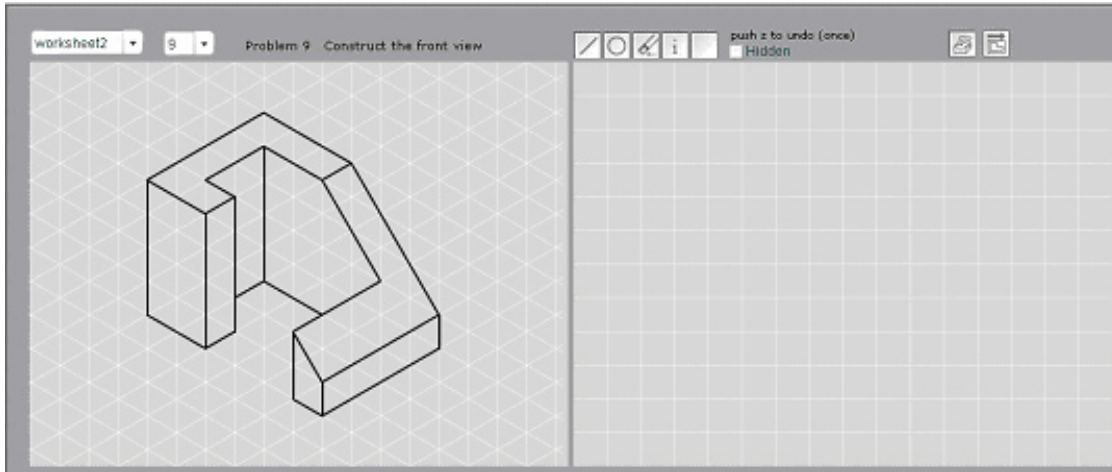
Finally, the tedious task of manual grading will be eliminated, since the program provides instantaneous performance feedback. This is a significant pedagogical advantage also, as the students benefit from the immediacy of feedback on their efforts.

Although the tutorial with its interactive drawing tool is still in the fundamental development stages, the authors are encouraged by the apparent potential of the product to significantly alter the way this topic is taught (and learned) in existing engineering graphic curricula.

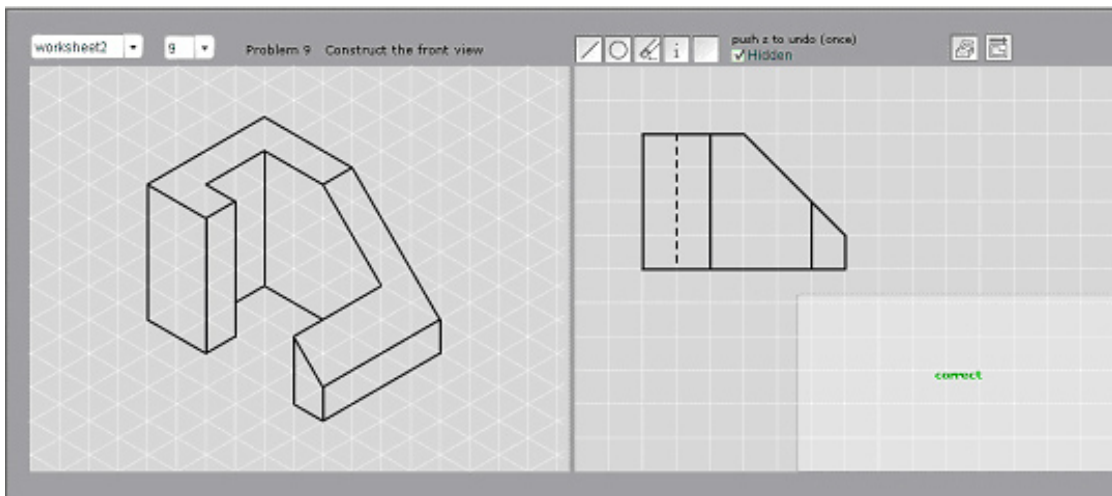
References

- Alessi, S. M., & Trollip, S. R. (2001). *Multimedia for learning: Methods and development* (3rd ed.). Boston : Allyn and Bacon.
- Bertoline, G. R., & Wiebe, E. N. (2003). *Technical graphics communications* (3rd ed.). Boston : McGraw Hill.
- Connolly, P. E., & Maicher, K. R.. (2003). Instructional design of a web-based engineering graphics tutorial. *Proceedings of the 58th Annual Engineering Design Graphics Midyear Meeting*, Scottsdale, Az., 89 - 97.
- Eom, W., & Reiser, R. A. (2000). The effects of self-regulation and instructional control on performance and motivation in computer-based instruction. *International Journal of Instructional Media*, 27(3), 247-260.
- Holliday-Darr, K., Blasko, D. G., & Dwyer, C. (2000). Improving cognitive visualization with a web based interactive assessment and training program. *The Engineering Design Graphics Journal*, 64(1), 4-9.
- Maicher, K. R., & Connolly, P. E. (2003). Using web authoring applications in tutorial development. *Proceedings of the 58th Annual Engineering Design Graphics Midyear Meeting*, Scottsdale, Az., 166 - 174.
- Newby, T. J., Stepich, D. A., Lehman, J. D., & Russell, J. D. (2000). *Instructional technology for teaching and learning*. (2nd ed.). New Jersey: Prentice-Hall.
- Poli, C., Fisher, D., Pollatsek, A., & Woolf, B. P. (2003). Design for stamping: Identifying pedagogically effective components in multimedia tutors and the classroom. *Journal of Engineering Education*, 92(3), 227-237.

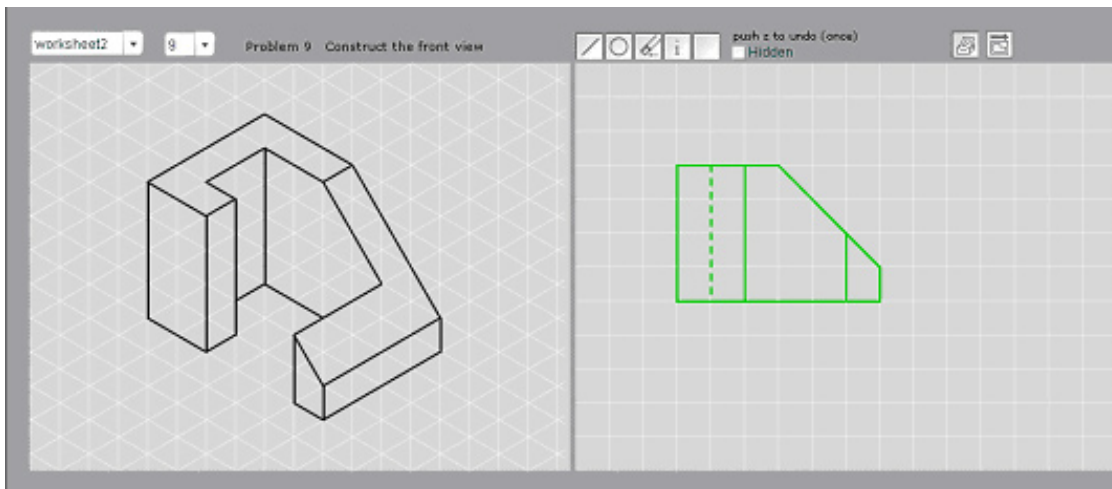
Appendix A. Interactive Drawing Tool Examples



Interactive Drawing Tool - Beginning of Problem

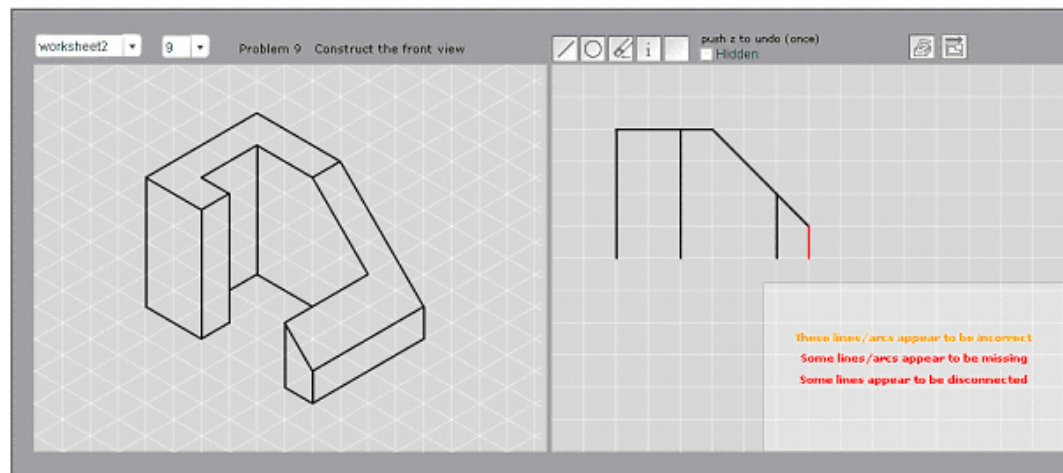
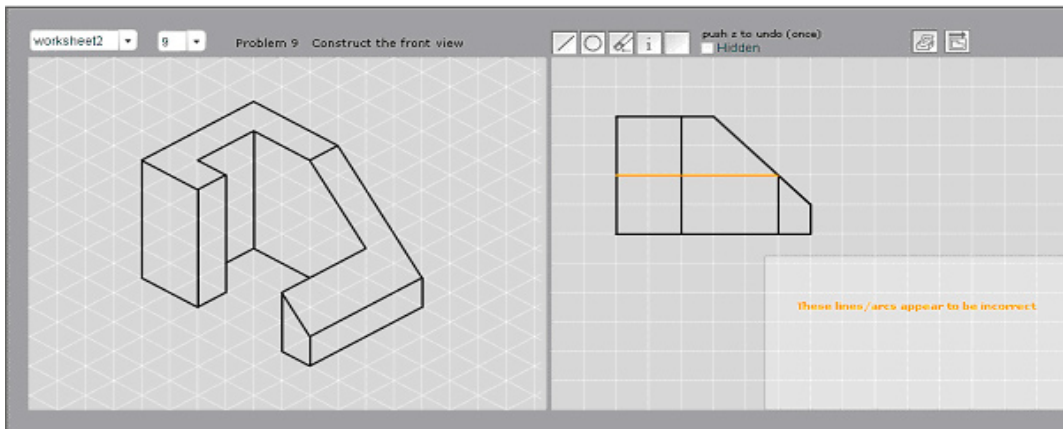
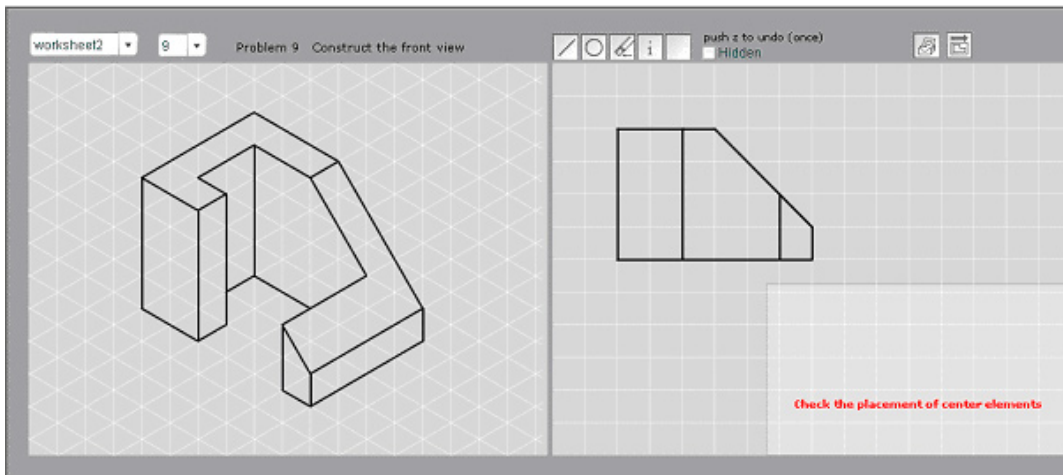


Interactive Drawing Tool - Successful Completion (Grade Drawing Option)



Interactive Drawing Tool - Get Answer Option

Interactive Drawing Tool - Sample Error Messages



Appendix B. Survey Questions

1. Was the product easy to use and understand?
2. Were the error messages clear and helpful?
3. Describe the errors you encountered.
4. What did you like about the tutorial?
5. What did you not like about the tutorial?
6. Would a tool like this be helpful in your understanding of multiview drawings?

7. What changes would you make to the product?

Acknowledgements

The authors would like to thank Johnny Slack and Evan Short for their outstanding efforts in the programming and endless adjustments to the interactive drawing and response tool.

An Investigation of Solid Modeling Practices in Industry

H. K. Ault and D. T. Giolas
Worcester Polytechnic Institute

Abstract

Parametric solid models provide a quick way of constructing parts that can be easily modified and redesigned for reuse in a variety of downstream applications. However, the method used to create the model has a significant impact on the level of usability. This research uses a combination of interviews, company standards evaluation, and model analysis to determine current industry practices for the creation of solid models. The focus is on creation of single part models. The paper includes a summary of differences in modeling methods based on designer preferences and software functionality.

Introduction

The goal of this study is to identify industry practices in solid modeling. A comparison of modeling practices used by experienced designers may be used to identify strategies that will improve or increase the usability of the solid part models and reduce the need for remodeling.

Models are typically started with a primitive or a sketched feature. Primitives can represent the stock material from which the parts can be machined. Sketched features produce geometry by extruding or revolving a 2D profile. The first primitive or the first sketched feature is usually located relative to a global Cartesian coordinate system. Additional reference entities, such as datum planes, axes, coordinate systems and points, are often created for constraining purposes. Subsequent features can be of any type and are constrained by references to existing part geometry. Adding the cosmetic features (e.g. fillets, chamfers, and threads) is typically the final step in the modeling process.

Background

The majority of solid modeling research, by far, is in the area of Product Data Management (PDM). This includes mostly database management, concurrent engineering and intelligent CAD (eg. Bronsvort and Jansen, 1993). A small area of research is based on designing for assembly, where tradeoffs are made between the number of assembled parts and the complexity of each part (eg. Kim, 1997; He, Kusiak, & Tseng 1998). Other work being done in the area of modeling is for the purpose of finite element analysis. This involves

the simplification of models to improve the efficiency and accuracy of the FEA (eg. Mathews, 2002; Rolf, 1995). Little or no work can be found in the literature regarding part-modeling strategies.

The majority of work dealing directly with the creation of individual solid models is presented at annual user conferences organized by the software companies themselves (Naujok, 2002). However, these presentations focus mainly on the functionality of the individual software package and not on the general practice of solid modeling itself. The research presented here investigates the current solid modeling methods used in industry and generalizes the practices that apply to all major solid-modeling programs.

Research Objectives

The Part-Modeling Process

For a part-modeling process to be efficient it must follow a structured methodology for solid modeling. The methods used to create solid models may differ greatly depending on the particular downstream applications, product type, software functionality, required documentation and designer preferences. Possible solid modeling methodologies include a fabrication approach where the choice of features is based on the actual manufacturing process, e.g. material removal. Another method includes focusing on the constructive building blocks or sculpting/shape manipulation, where models are created using the simplest geometries to make analysis easier. This paper will use a combination of interviews and part structure analysis to capture some of the preferred

List of Interviews				
Interview	Job	Company	Software	Part Types
1	Designer	A	Pro/Engineer	Machine Parts
2	Designer	B	Pro/Engineer	Machine Parts
3	Designer	C	Unigraphics	Machine Parts
4	Designer	D	SDRC IDEAs	Mach./Molded Plastic Parts
5	Designer	D	Unigraphics	Car Audio Parts
6	Designer	E	AutoDesk Inventor	Computer Cases
7	Designer	F	Catia	Tooling
8	Standards	G	Pro/Engineer	-

Table 1

methods or best practices for the creation of solid models used in industry. A similar technique was used by Flemming, Bhavnani, & John (1997) in an investigation of practices used for 2D CAD (computer-based drafting).

Interviews of Model Designers and CAD Staff

The majority of the interviews were conducted with experienced mechanical designers, both engineers and CAD designers. All of these people used a solid modeling program on a daily basis. These individuals employed a variety of solid modeling programs. The level of experience ranged from those who started their design careers on drafting boards to those who started solid modeling right out of college. The parts being designed by this group included a combination of machine parts and product parts. Table 1 lists the interviews with company, jobs, software packages and type of parts modeled.

Study of Company Standards

Requests were made of the companies interviewed for any company standards regarding solid modeling. These standards only existed at the larger corporations, and company policies prohibited them from providing copies of the official documents. We typically obtained copies of the presentations used to present the standards to the designers. These documents demonstrated that the standards focused mainly on Product Data Management (PDM) and not on solid modeling best practices. However, some interesting results came from those standards that were collected, and are presented here.

Evaluation of Existing Parts

Files of designed parts were obtained from most of the designers interviewed. The standard

practices used by that designer were extracted from these parts. The parts were evaluated using the native program to determine the particular method used to create the solid model. The model tree of each part was evaluated to determine the order of feature creation and the choices made for the constraint of each feature. Any specific profiles created by the user were studied to determine the design intent suggested by the specific parameters chosen. Finally, the results from the models were compared to the 2D drawings for similarities and differences in dimensioning schemes and associativity with the solid model.

Evaluation of a Common Part

Due to the wide variety of parts received for evaluation, a direct comparison of modeling styles proved difficult. The decision was made to have a number of designers each create a model of the same part. The part was chosen from the group of designs received. One part received from Company B seemed to use the widest variety of geometry and more choices for constraints were available to the designer. An orthographic drawing for the part was created from scratch, shown in Figure 1, using the same major features, eliminating any details referring to proprietary information. The simplified geometry and randomly chosen integer dimensions provided similar information without requiring a prohibitive amount of time from each designer. The drawing provided only the critical dimensions; all other geometry was to be scaled. The designers were asked to model the common part and return the native files to the research team. The part files were then studied to determine the modeling methods used by the designers.

Discussion

Interviews

The results of the interviews with the

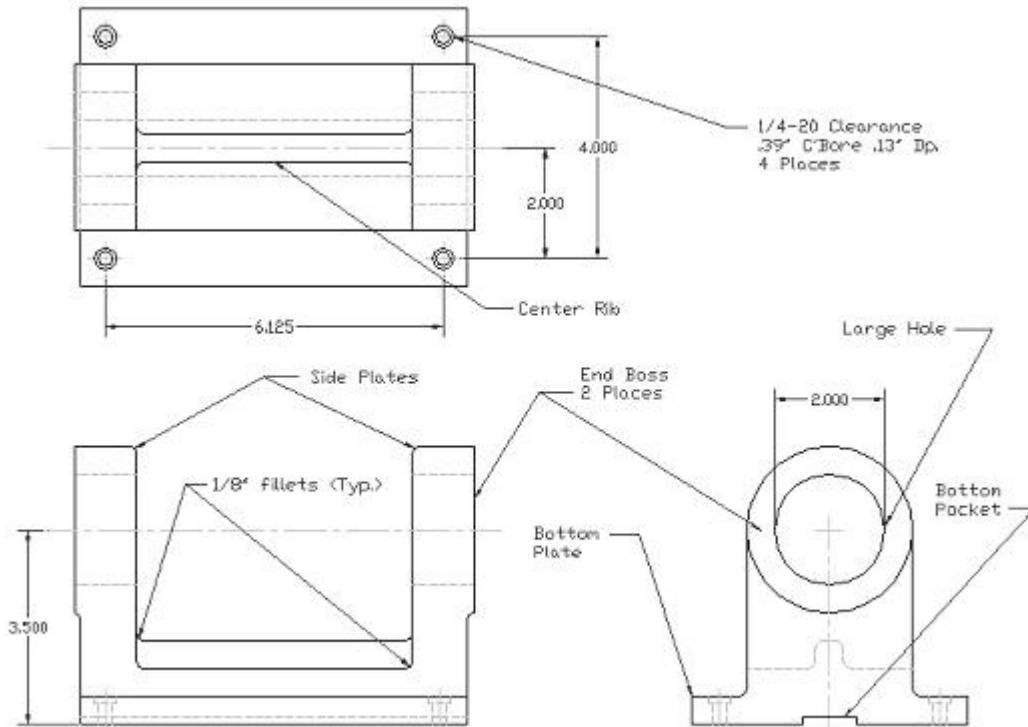


Figure 1 Drawing of Common Part

designers proved the most fruitful. A number of similarities were found and some unexpected differences. Some of these differences appeared to be the result of differences in software functionality. However, some differences in modeling strategies or methods were also found. The main similarities came in the form of general rules of thumb.

Designer Similarities

The most important similarity, also the most general, involves planning. Everyone agreed on the importance of doing some planning before any geometry is created. The purpose of planning is to determine the critical dimensions and references to provide for easy modification. Proper references allow for capturing the designer's intent, and facilitating future redesigns. Interestingly, the designers who had started on drafting boards recommended the part should be sketched on paper before modeling, but some others used the modeling program as a planning tool to quickly sketch out ideas. Upon completion of planning, it was not uncommon to start the final model from scratch. The process of designing versus strictly modeling produced different results. Designing is

a dynamic process that usually changes direction a number of times before completion, creating a disorganized model. The model creation, after most of the design decisions are made, is quick and produces the cleanest model. However, this can never be done completely, and rarely does a designer look back at a complicated model and not wish something had been done differently.

Of those interviewed the consensus when selecting references was to define features with the same constraints and dimensions as those used to generate the corresponding 2D drawing. The selection of dimensions contained a mixture of goals. One common method is to design parts in a way that is consistent with the manufacturing process. The second, but equally important, is to prepare for the features and dimensions that are most likely to be modified. References and parameters are then selected to represent these dimensions as simply and logically as possible.

Parts are typically designed to mate with other parts. When parts are designed in an assembly mode, the parts use each other as references to ensure that they fit together. Parts that reference other parts are said to have external references.

Linking parts is a powerful tool; it facilitates the design of parts and reduces the chances of making errors while manually transferring data between parts. However, it was generally agreed that the links between individual parts should not be maintained beyond the initial design process. Once the model is finalized, the part should not change accidentally due to a modification of some other part. The software companies supply this functionality to allow for a continually self-updating model. However, most designers feel that these links are too difficult to maintain on a continuous level. External references are still used to help define the datums and matching geometry. However, these links are typically removed prior to finalizing the model, requiring some level of redefining.

Designer Dissimilarities

Some interesting differences were noted. Designers expressed different opinions about the use of sketches. Some expressed a preference to extrude or revolve complicated sketches in order to reduce the overall number of features that comprise a model. Other designers preferred to keep the sketches relatively simple and use additional features to create the complicated geometry. The latter group felt that simpler sketches and features produced a more manageable model. The software programs contain numerous tools, such as model trees, to manage features and how they relate to each other. Yet, complicated sketches do allow for easier relations to be made between parameters. The former group felt that the sketches communicate the majority of the design intent, and they strived to keep the model tree as simple as possible. These differences may be only a matter of different personal styles, but the differences are significant.

Differences also existed on the use of chamfers and rounds. Most users suggested chamfers and fillets should be applied towards the end of the model. A few felt this could cause errors with previously generated features. These problems are most likely a function of the software's method of handling these features rather than the method. Pro/Engineer treats chamfers, rounds and drafts as separate features that do not affect references of the existing geometry. Unigraphics, on the other hand, operates differently. Chamfers, rounds and drafts modify the identities of the surfaces

they affect causing errors for other features referenced to that surface. For this reason, users of Unigraphics generally recommended that all geometry be referenced from separately created datum planes instead of from the surfaces of existing features. Chamfers, rounds and drafts are generally cosmetic and have minor importance to the main functionality of the part. It stands to reason that they should be added at the end of a model wherever possible.

Company Standards Standards Focus on Documentation

The lack of official company standards suggests that most of the companies are interested, but have only begun to think about it. A few companies have been working on a set of standards but most of those were still in the rough draft stage or were created in house and not yet made official by the company. Due to the unofficial nature of these standards and due to proprietary issues, most companies were unwilling to show what they had produced so far. The standards found were not specific enough to solid modeling methods and generally covered more bookkeeping methods, such as colors, layers and file-naming conventions.

Company G Standards

Company G was unwilling to show us any official documentation, but instead provided the presentations used to explain the standards to the designers. The practices covered in these presentations focused mainly on file usage, program and system performance. Their largest concern seems to be concurrent engineering and database management, but the standards do list general practices and common problems. The standards relating directly to modeling involve the use of features. Unigraphics allows for the use of primitives, as well as predefined features and profile-defined features. Company G recommends the use of features wherever possible. Mixing primitives and features can cause dimensioning problems when creating drawings. Recommended practices include:

- Assembly cuts should be avoided especially cuts that remove features.
- Holes should be generated with the Hole tool instead of as a cut.

- Hole patterns should be used wherever possible.
- Excessive use of model cuts should be avoided.
- Simple profiles are preferred over complicated sketches.
- Dimensions should be driven by the model, not applied on the drawing as unassociated markups.

Company C Standards

Company C has produced a three-page document describing standards encompassing all CAD work. About one page of this covers the actual creation of solid models. The majority of this document covers bookkeeping practices, such as file management, format preferences, and preferences for units, start parts, layers and drafting settings. The section concerning solid modeling contained several general practices, some relating directly to Unigraphics. These include:

- All parts should be fully constrained on a coordinate system.
- Tolerance settings and body check should be used to ensure the geometry has no discontinuities.
- Final parts should be subtracted from a solid to check for cavity creation.
- Feature arrays should be used whenever creating a pattern of features.
- The copying or splitting of solids should be avoided.
- All features should be placed with positioning dimensions.
- Face offsets should be avoided.
- Symmetrical objects should be mirrored.
- The final part should not contain any suppressed objects.

The majority of these recommendations are specific to the functionality of Unigraphics. Unigraphics relabels surfaces when they are modified which causes linking problems with existing features that are referenced to these modified surfaces. The copying, splitting and offsetting of features is not a problem in other software packages, such as Pro/Engineer and Solidworks. However, fully constraining features, feature arrays and mirroring features seem consistent with most modeling practices.

Company D Standards

Company D has produced the most detailed documents regarding modeling strategies, including graphical examples. All of these standards are collected in an intranet website to provide easy access to the specific aspect required. The three main sections of this site are Part and Assembly File Structures, Standard Component Modeling and Non-Standard Component modeling. The latter two have graphically demonstrated examples. This website includes a list of general modeling practices with explanations of why these practices are superior. These explanations seem to be a key component in getting "buy in" from designers who have been doing it "their way" for a long time. These standards were created by the employees and have not been company mandated.

The Modeling Guide, contained in the website, discusses the advantages and disadvantages to the use of sketched features versus predefined features. Sketches are very powerful for creating complicated profiles and can easily convey design intent. However, sketches and their datums tend to clutter models and require more layers to maintain good file structure. Form features are a quick and easy way to generate simple geometry and can be easier to edit than some sketched features. However, form features have to be located explicitly to the adjoining solid. These connections can make it difficult to change their positions later. Primitives may be used to create the starting feature of a part but their use is not recommended beyond that. Other areas identified for improvement include the use of spreadsheets to control and edit expressions and the idea of creating and adding user-defined features.

In addition, Company D has created a number of generic part files representing some of the common geometry used in their applications. These parts, called Seed Parts, can be easily modified for any application, and are maintained in a central location so they can be modified to take advantage of any future improvement in modeling strategies. Company D has recorded significant reductions in modeling time for standard components when the appropriate Seed Part is used. The time to model a particular part was reduced from 16 hours to 4 hours and the resulting part was significantly easier to modify than the previously created models. Some employees felt that the Seed Parts provided

more educational value than the instructional website itself.

Company D is clearly ahead of the other companies interviewed. Most companies are just beginning to consider standards. Company B has created an outline but is only beginning to write the document. Company F has a rough draft in process but was unable to show any of it. However, they did mention a central modeling lab where designers can be trained and the latest best practices demonstrated. Clearly, solid modeling best practices is still a young topic in industry. For true modeling consistency, company standards should generally require a more detailed description of modeling practices and contain demonstrations of case studies.

Existing Parts Wide Range of Results

Most of the people were reluctant to provide real parts to analyze. However, the parts received varied in type and shape, due to the fact that different industries were included in this study. From five companies, three designers donated parts. A total of nine parts were received; four were modeled using Pro/Engineer and five were modeled using Unigraphics. The results demonstrated a number of modeling styles. However, this variation might be considered a function of the various types of parts. In a number of cases, complicated profiles were used to create the initial geometry, and at the same time, it was also common to start with a simple shape and then add various profile cuts to produce the complex geometry. Finally, except for only a few departures, the critical features used to dimension the drawing were the surfaces used as the constraints in creating the model.

Part History's Effect on Design Intent

We were able to check the history of each model. The original expectations were that the older parts with a complicated history contained less intuitive design intent. However, no data were found to support this hypothesis. Unfortunately, most of the parts did not have a particularly complicated history. Most of the entries into the history tree represented various revisions of the start part. All but two of the parts had been created and modified by the same designer. The two

models with two or more designers contributing to them were two of the better-designed models. This was not a large enough sample size to make a conclusion.

The Common Part

The focus of this trial was on the order and method use for feature creation. To determine the usability of each model, a number of changes were performed and evaluated for ease of modification. The greater number of dimensions changed or features redefined during each modification lowered the usability of each model. The two models evaluated varied greatly in style. The number of responses received was limited due to time constraints. Only two were readable with available software.

The first response, which was created in AutoDesk Inventor 7, had most of the geometry created with the extrusion of the first sketch. The second sketch was used to cut out the center section creating the two side supports and the center rib. Two datum planes, offset from the center plane, were used to create the boss on each end and the sketches were based off of the existing geometry. A third datum plane, based on the top surface of the bottom plate, was used to sketch and extrude the four holes. The majority of effort was put into the initial two sketches, a very efficient method.

The sketch to create the first feature, shown in Figure 2, was very intuitive and allowed for easy modification of the most of the dimensions. During the evaluation changes were made one at a time to the height and diameter of the large hole, the length of the part, and the location of the counterbored holes. The height and diameter were easy to change within the first sketch. One problem appeared when the length was changed. The two end-bosses were linked to two independent workplanes and not to the exiting geometry. This caused the bosses to remain in position while the other geometry moved. When questioned, the designer explained that he prefers to sketch on planes rather than on geometry surfaces. Sketches require more computer time to regenerate and the geometry gets regenerated more often than the datum planes.

Another complication arose when modifying the location of the counterbored holes. An extra

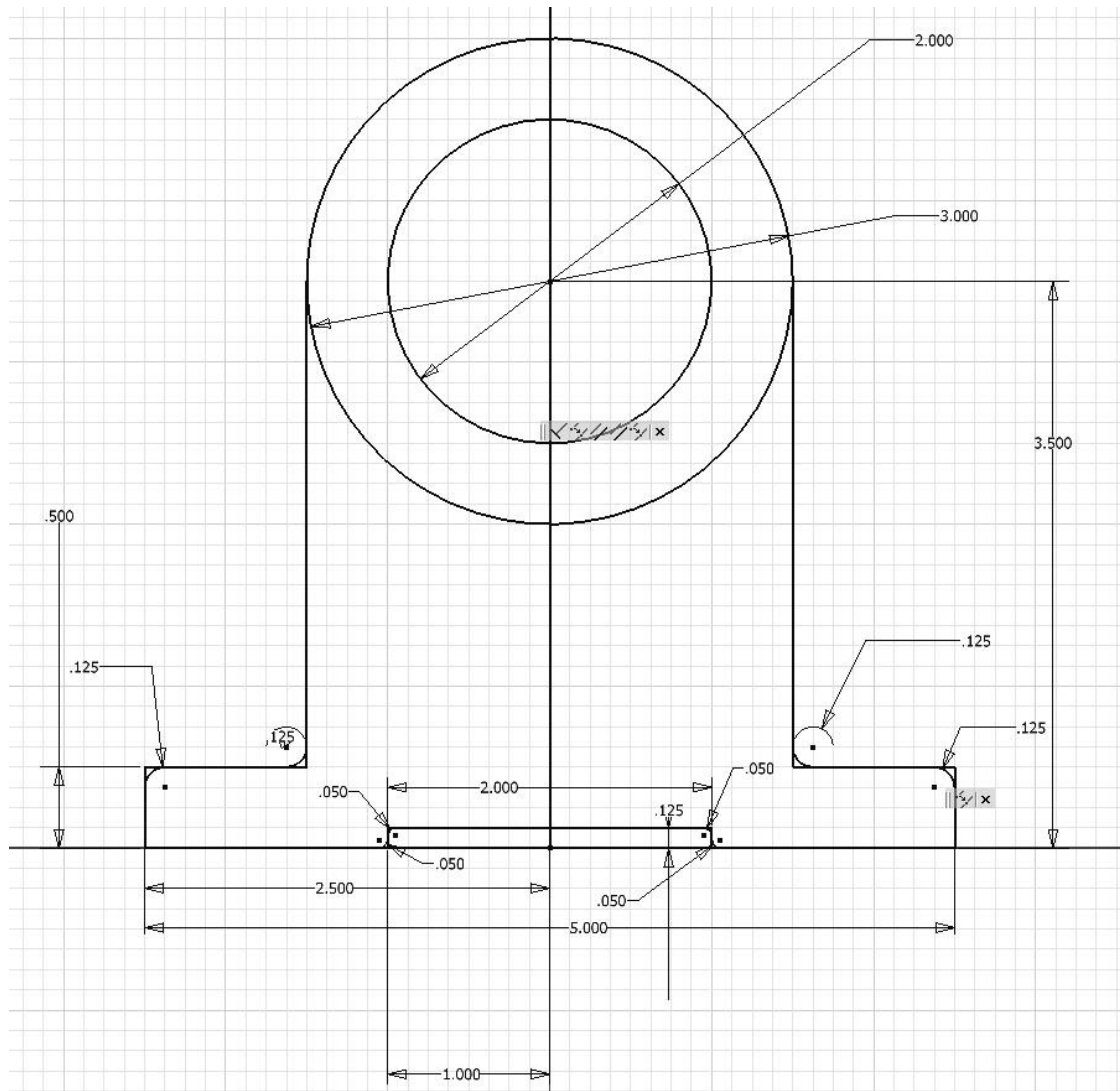


Figure 2. Initial Sketch for Common Part modeled in AutoDesk7

plane was created by the designer, so a problem would occur if the thickness had to be changed. The holes would stay with the plane and not move with the geometry. In addition, the Hole command wasn't used. Two sketches were used, one for the through hole and one for the counterbore. Changing the location of the hole would require the modification of two separate sketches in order to move the holes and since the patterning tool wasn't used each hole would need to be moved separately. The creation of the main geometry with two sketches worked well, but the creation of the secondary geometry (e.g. end bosses, holes) seemed to follow an "easy to model but hard to modify" methodology.

The second part based on Figure 1 was created by a different designer in Unigraphics, and con-

tained more setup planning. It was built from the ground up. The order of geometry creation included the bottom two rectangular plates, the center rib, the two side plates, the counterbored holes, the two endbosses and the large hole was last. The large hole was kept on the axis of the side blends by the use of a mathematical relation. This relation initially made modifying the height of the hole difficult, until the table containing all the parameters was studied to understand it better. The counterbored holes were made with the Hole and Patterning tools, making them easy to modify. The length was also modified without problems. The largest problem was that the heights of the side plates, clearly a critical dimension, were referenced from the top of the first plate and not the bottom. Using this dimension would be difficult when modifying the part. Overall, this part

Summary of Results					
Company	CAD System	Fully Constrained	Primitives	Sketches	Fillets
A	Pro/Engineer	Yes	No	Complicated	End
B	Pro/Engineer	Yes	No	Simple	End
C	Unigraphics	Usually	Yes	Simple	End
D	IDEAs	Yes	No	Complicated	During
E	AutoDesk Inventor	Yes	No	Complicated	End
F	Catia	Usually	No	Complicated	End
G	Pro/Engineer	Yes	No	Simple	End
H	SolidWorks	Usually	No	N/A	End
I	Pro/Engineer	Yes	No	Simple	End

Table 2

might have taken a little longer to create but proved easier to modify.

In comparison, the second model would seem superior. However, with a few modifications, the first would prove to be faster to create and easier to modify. The sketch used to create the main geometry was easier to understand and focused more on the critical dimensions. The recommended changes to the first method would include sketching the end-bosses directly onto the model surface and using the standard Hole and Patterning tools to create the counterbored holes. Essentially, this would be a combination of the two styles discussed here.

Conclusions

Consistent General Rules

Upon final consideration of all of the interviews and sample parts, the solid modeling methods used by various designers had much more in common than originally expected. The results are summarized in Table 2. The main list of common practices is as follows:

- All parts and features should be fully constrained. Despite some software ability to provide it, partially constrained geometry is generally not recommended.
- Simple or sketched features should be used wherever possible. Primitives are rarely used and, even then, only as the initial geometry of a part. Features provide more options for constraining and parameterization. Mixing features and primitives can cause different types of problems, depending on the particular software.
- Cosmetic features, such as fillets, rounds and chamfers, should be applied towards the end of the model and not have dependents. Suppressing these features to create a simplified representation is often required for downstream applications, such as FEA and large assemblies.

A wide variety of industries were sampled here. The parts designed in the industries sampled included machine parts, small molded/machined plastic parts, tooling and computer cabinets. At least six different modeling programs are being used throughout these industries. The differences in style may be a function of the type of part being modeled, as well as the software used.

Differences Between Designers

A few inconsistencies exist that seem to be due to personal preferences. The desire to use one complicated sketch versus a number of simpler sketches tends to vary from designer to designer. Some feel that the sketch provides the best method of communicating design intent. Creating all of the parameters in a feature eases the use of algebraic relations to control the geometry. The goal of using a complicated sketch is to keep the model tree as simple as possible. Others feel that simpler sketches are easier to manage and prefer to use the model tree to display the relationships among the different features.

Many designers also prefer to design using a material removal or machining methodology. This generally involves starting with the simple initial shape and using cuts to produce the desired parts. This method easily lends itself to designing for manufacture. It also encourages the use of the model dimensions as the drawing dimensions. Others feel that the modeling constraints and drawing dimensions are rarely the same and that the modeling parameters often need to change based on the downstream applications, e.g. NC, FEA.

Variations Related to Software

Some of the differences in modeling strategies can be attributed to the particular software's functionality. Occasionally, it was suggested that

fillets and chamfers should be added throughout the process instead of at the end. The reason is that these features often cause errors if other geometry is referenced off of the surfaces affected by that chamfer or fillet. These designers also preferred to reference all features off of datum planes and not off of existing surfaces. This problem is a software issue with how Unigraphics identifies each surface and then re-labels it after being modified, causing problems with anything dependent on it. This is not a problem in a number of the other solid modeling programs available.

Not all designers inherit the model dimensions into the drawing. The dimensions in the inherited form are often too complicated to be usable. Designers using Pro/Engineer believe this to be a good practice and generally design this way. This suggests that inheriting dimensions is possible with the proper software functionality, although the type and complexity of part designed may affect the practice of inheriting dimensions.

Separate Designing from Modeling

The quality of a model is highly dependent on the amount of preplanning allowed. The situation producing the optimum results involves completely designing the part prior to the start of the modeling. Reproducing an existing design as a solid model can take, in many cases, less than an hour, whereas the original design may have taken days. Designing a part is a highly dynamic process and usually produces a less than perfect model. Decisions often change the model beyond the designer's ability to redefine the model in order to fully compensate for the new design intent.

Use CAD to Plan, Design and Model

Experienced designers recommend that the part be sketched out on paper before any geometry is created in the modeling program. This, however, goes back to the Flemming et al's(1997) T-Square metaphor where traditional methods are maintained for comfort rather than efficiency. Feature-based, parametric solid modeling can be an extremely powerful design tool when used properly. The designer needs to utilize these programs to plan and design the part. The easy addition and subtraction of features and the quick variations allowed by a parametric modeler can be used to plan the part and investigate as many

design variations as needed. However, for the sake of the future usability of these models, the final model should be kept separate from the numerous twists and turns to the design process. In other words, the model should be fully redefined to reflect the final design intent or, if necessary, recreated from scratch. This will often require extra time, but will produce a better design and a more robust solid model.

Future Work

This article notes some interesting differences in modeling strategy. The next step would be to evaluate the efficiency of each strategy. But first, the harder task of defining and measuring efficiency in a meaningful way is required. The clearest difference between designers is the use of one complex sketch compared with the use of a group of simpler shapes. Therefore, the use of sketches should be the first subject of an efficiency study.

This article compared the results of a several designers modeling the same part, and produced some interesting results. However, a more complete trial should include different types of parts and separate out the differences between industries, software and strategies. We hypothesized that many differences in the results were due to the functionality of each software package, but there are also differences attributed to designer preferences and company practices. We are reminded of the "shoot-outs" that were often organized at conferences to demonstrate the capabilities of software packages in the early days of CAD. Similar studies could be organized to compare solid modeling software.

References

- Bronsvort, W.F. and Jansen, F.W., (1993). Feature modelling and conversion--key concepts for concurrent engineering, *Computers in Industry*, 21, 61-86.
- Flemming, U., Bhavnani, S. and John, B. E. (1997) Mismatched metaphor: User vs. systems model in computer-aided drafting. *Design Studies*, 18, 349-368.

- He, D., Kusiak, A., Tseng, T.L., (1998). Delayed product differentiation: a design and manufacturing perspective, *Computer-Aided Design*, 30(2), 105-113.
- Kim, G. J., (1997). Case-based design for assembly, *Computer-Aided Design*, 29(7), 497-506.
- Mathews, R., (2002). Solid Modeling for FEA, 2002 Solidworks World Conference Proceedings.
- Naujok, C.E., (2002). Optimizing Solidworks through Standards, 2002 Solidworks World Conference Proceedings.
- Rolph, W. D., (1995). Requirements for Finite Element Model Generation From CAD Data — An Approach Using Numerical Conformal Mapping, *Computer & Structures*, 56(2/3) 515-522.