AC 2010-138: STRATEGIES FOR TEACHING CAD AUTOMATION TO ENGINEERS AND TECHNOLOGISTS

Derek Yip-Hoi, Western Washington University

Derek Yip-Hoi is an Assistant Professor in the Department of Engineering Technology at Western Washington University and coordinator of the department's CAD/CAM program. He received his Ph.D. in Mechanical Engineering from the University of Michigan in Ann Arbor where he worked for several years as a Research Scientist in the area of Reconfigurable Manufacturing before moving out to the Pacific Northwest where he spent 3 years at the University of British Columbia before moving to WWU. His teaching interests are in CAD/CAM, CNC, design methodology, mechanical design and manufacturing processes.

Strategies for Teaching CAD Automation to Engineers and Technologists

Abstract

Training in Computer-Aided Design is now common place in engineering and technology programs. This can take one of three forms. Instruction in the mechanics and strategies for effectively using a CAD application is the most common. This is often completed early in the program to provide students with the ability to use these tools in term projects and capstone design. The second approach delves into a study of the building blocks of a CAD system getting into the areas of geometric and solid modeling, constraint solving, data structures, computer graphics and the use of CAD data in downstream processes such as tool path generation and rapid prototyping. Courses in this area are typically offered as senior electives or in graduate programs. The third form involves teaching how to automate CAD system functions using simple programs commonly referred to as *macros* or *scripts*. In the past CAD system vendors have provided their own scripting languages. Examples include AUTOLisp (AutoCAD) and GRIP (Unigraphics). Today with the use of Windows building blocks in CAD interface development, Visual Basic for Applications (VBA) is now commonly available for scripting (e.g. Inventor, SolidWorks, CATIA, SolidEdge to name a few). Study of this area which is referred to as *CAD Automation* in this paper is also at the senior level or in a graduate course.

This paper describes in detail a new senior level course being offered to Engineering Technologists that teaches CAD Automation using CATIA with VBA. Students are introduced to the various areas in which CAD automation can be applied. These are in automating the activities of *Part Configuration*, *Product Configuration*, *Integration*, *Data Retrieval* and *Analysis and Optimization*. Strategies and techniques for teaching the automation tools are presented. This is challenging in part because of the variation in programming backgrounds of the students. The use of *Excel* a program that students are familiar with, to teach VBA is described. It will be shown how this is also critical to the study of the *Integration* aspect of automation where data is passed back and forth between a spreadsheet and the CAD system in creating and manipulating geometry and product structures. Teaching the CATIA automation object structure is accomplished through study of macros recorded during manual modeling activities with the GUI. The benefits and challenges in using this approach are discussed.

Overviews of assignments and project work are given. Assignments include the creation of a beam bending program in *Excel* that controls beam section and length parameters in CATIA and that extracts section properties in calculating beam deflection and stresses. An example of project work that involves automating the creation of airfoil sections for products such as aircraft wings, propellers, helicopter rotor blades, wind turbines and hydrofoils is described.

This paper concludes with a discussion of the challenges observed in teaching this course and ways to improve content and delivery in the future.

Introduction

The Engineering Technology Department at Western Washington University (WWU) runs several programs that train Technologists in the area of product development. These include

ABET accredited programs in Plastics and Manufacturing Engineering Technology (ET) and Industrial Technology (IT) programs in CAD/CAM and Vehicle Design. As with all technology programs a focus on hands-on project work to supplement rigorous coursework is considered critical to a well trained technologist upon graduation. Training in CAD is considered an essential component of this. Traditionally this has taken the form of teaching the fundamentals of using a CAD system to create 3D models and engineering drawings. Two freshman introductory courses in Engineering Design and Graphics are used for this purpose. Students in the IT-CAD/CAM program are required to take two additional CAD courses that cover more advanced topics. These courses are in Assembly Design and Mechanisms Modeling, and in Surface Design and Modeling. These provide specialized skills which help to broaden the experience of CAD/CAM students. They are in keeping with the types of modules currently bundled in standard CAD systems that a specialist in this field might be expected to use upon graduation. CAD/CAM majors are also required to take courses that teach both manual part programming for CNC operation and the use of a CAM application to automate the generation of NC tool paths. The primary CAD system of use at WWU is CATIA (currently V5R19). The education bundle provides integrated workbenches for Kinematics, Generative Shape Design and Prismatic and Surface Machining. This has the advantage of allowing students to work seamlessly on models in the CAD and CAM environments without the need for translation while maintaining associativity between the data generated in the different workbenches. Pro/Engineer has also been used to teach advanced Assembly Design and Mechanisms Modeling.

CAD Automation

The courses described above train technologists in building models using the Graphical User Interface (GUI) of the CAD system. Students are taught how to use intelligent parametric modeling to capture *Design Intent*. This facilitates to some extent the ability to respond to engineering change orders or to create a variant of a model. Specialized techniques such as *Design Tables* can also be used for this. However, more involved part and product configuration changes may not easily be made using these techniques. In addition, the use of customized application interfaces that can speed-up execution of a change, reduce the likelihood of an error in input and make the generation of complicated models more accessible to a novice user are desirable. Consider the design of an airfoil. The cross-section of an airfoil is governed by parametric equations based on form generation standards (e.g. NACA 4, 6, 8 digit series). These are modeled by fitting spline curves through points generated from these equations. It would be incredibly tedious and error prone to have to input coordinates for each point for each section of an airfoil using the GUI in building its 3D model. Changes to the model would likewise be difficult to make. The use of a programming interface to the CAD system to accomplish this functionality is an appropriate solution and is what is referred to in this paper as *CAD Automation*. The programs developed using this technique are commonly called *macros* or *scripts*.

Macro programming is one of several automation technologies that are now commonly packaged with CAD systems. These can be broadly classified into three types:

- *Kernel APIs:* A kernel refers to a core functionality embedded within a CAD system. Examples include the geometric/solid modeler used to build and manage the 3D mathematical model of a part, the assembly modeler used to build and manage the constraints (mate, coincidence, align etc.) used in connecting components together and the sketcher's constraint solver used to determine if the geometry, dimensions and geometric constraints (tangency, perpendicular, parallel etc.) are consistent. Using the first as an example, ACIS[1,2], and Parasolid[3] are two commercially available kernels that are used in several major CAD systems (Inventor, SolidWorks, SolidEdge, Unigraphics). The kernel functionality is made available through libraries of APIs the functions from which can be called in programs written in a high-level language such as C, C++ or C#. These kernels are commonly used in courses that study the building blocks of a CAD system. Courses in this area are typically offered as senior electives or in graduate engineering programs. They require a sound grounding in computer programming and can have a high analytical content. Examples of texts that might be used in such courses include [4,5] and [6].

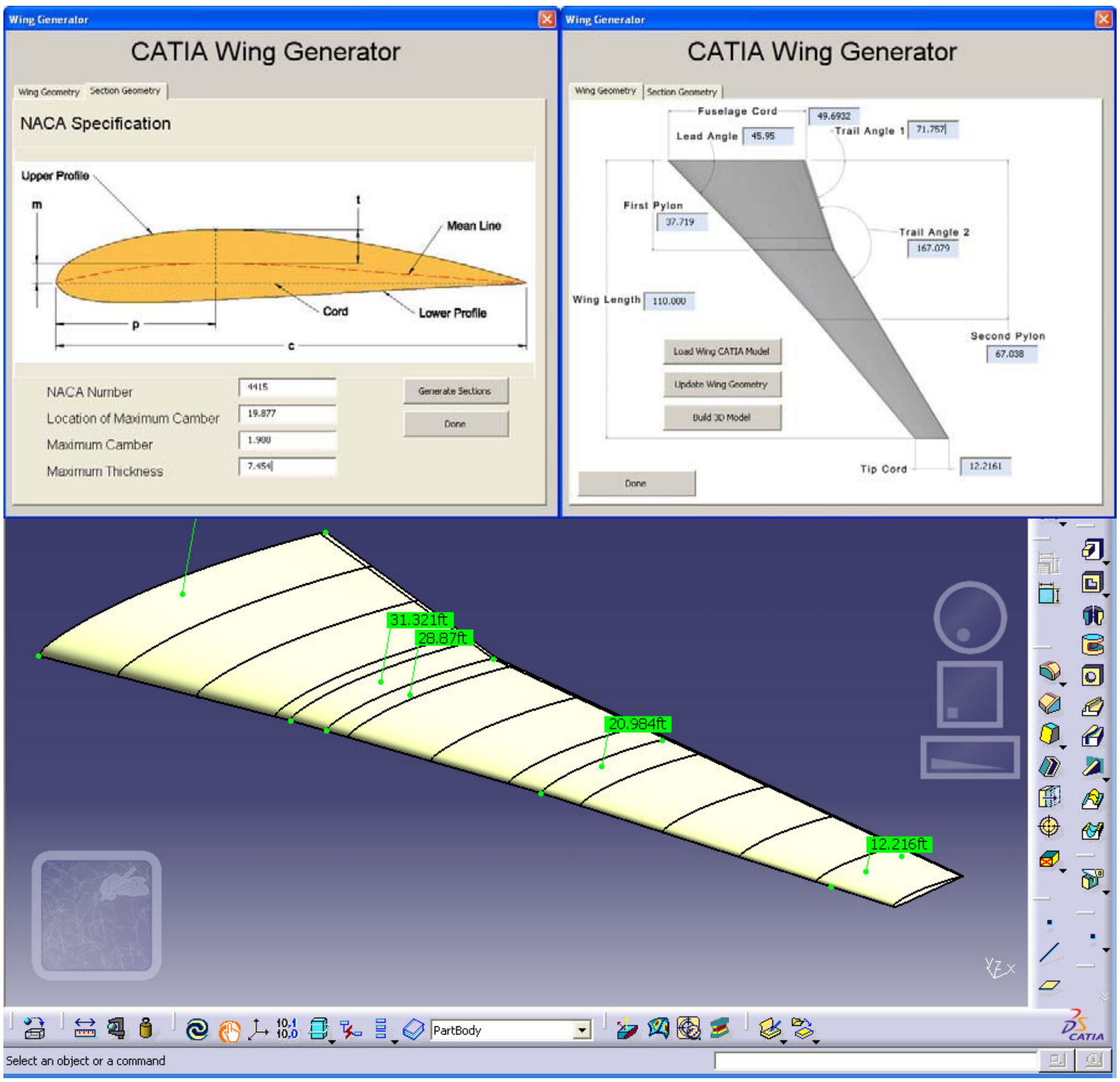


Figure 1. Automation Interface for Modeling of Wing using NACA Section Forms

- *Macros:* These are scripted instructions that are executed in interpretive fashion (no code compiling is necessary) by the process that runs in memory when the CAD system is launched. Many CAD systems have moved away from (or supplemented) their proprietary macro programming languages with VBA. One of the main advantages of using VBA is that integration with other applications that utilize the Windows object-based framework is simplified. Macro programming typically provides a means to programmatically execute the same operations that can be performed using the GUI. Access to the underlying kernels if at all possible is cumbersome. Macros execute more slowly than compiled code that run as independent processes. Texts for macro programming using VBA for different CAD systems are available most commonly for SolidWorks[8] and AutoCAD[9].

- *Knowledge-Based:* These automation tools have been developed to support the development of *Knowledge-Based Engineering* (KBE) applications within the CAD environment. Examples include *Knowledge Fusion* in Unigraphics and *Knowledgeware* in CATIA. Engineering design processes can be captured using these tools leading to the reduction of redesign effort by executing these processes for new designs. While macros emulate interactions with the CAD system to accomplish automation, these tools emulate engineering process steps that are related to creating geometry or making changes to the underlying CAD model. KBE tools are currently embedded in the larger CAD systems (UG, CATIS, Pro/Engineer). Training through the vendor or their supporting education houses is required for these tools.

Of the three techniques described the use of macros is best suited for exposure of technologists to CAD automation. The use of kernels is complicated by the programming skill that is required and the need to understand the foundational theory e.g. geometric and solid modeling. Knowledge-based techniques can be programmatically challenging particularly if the language is non-procedural (e.g. AutoLISP, Knowledge Fusion). These environments can also be so specialized that it unlikely a technologist will have access to it in a work setting after graduation. For example, a C++ compiler required for programming with kernel APIs is not standard fare on a CAD/CAM engineer's workstation in a small company.

An elective course ETEC 461 has been introduced at WWU to teach CAD Automation to CAD/CAM majors using the macro approach. This course uses CATIA, Excel and Visual Basic for Applications (VBA). The choice of VBA as the development environment builds upon the student's programming experience acquired in their required Computer Science course. It can also be viewed as an accessible language as several CAD systems have moved away from their own proprietary macro language to integrate this as a standard programming interface. The use of Excel provides an easy to understand case study to introduce the concept of Object Oriented Programming and is critical to CAD automation since much of the data that might be used to drive an automated model (the airfoil example) is easily generated on a spreadsheet. Exposure to VBA programming in Excel is also a highly transferable skill to non-CAD problems e.g. machine design, manufacturing costing and quality control. It is an accessible tool to many technologists through the Office suite.

The remainder of this paper will describe this course. The next two sections will highlight the teaching strategies adopted. These will be followed by sections that provide examples of assignments and projects used to develop a student's skill at developing automation applications. Finally summaries of challenges and potential future improvements to this course will be provided.

Course Overview

Since WWU operates on the quarter system, courses are scheduled over a 10 week period. As a four credit offering the *CAD Automation* class meets for two 3 hour periods in the department's CAD laboratory. The size of the lab caps enrollment at 25 students. This adequately meets the demand for the CAD/CAM program while providing space for students in other programs wishing to take this course as an elective. The course objectives as presented to the students are as follows:

- To introduce students to CAD automation techniques used to create and manipulate 3D parametric models
- To deepen a student's understanding of the structure and uses of 3D parametric models
- To develop a student's skill at using programming tools to solve engineering problems in design and manufacture.

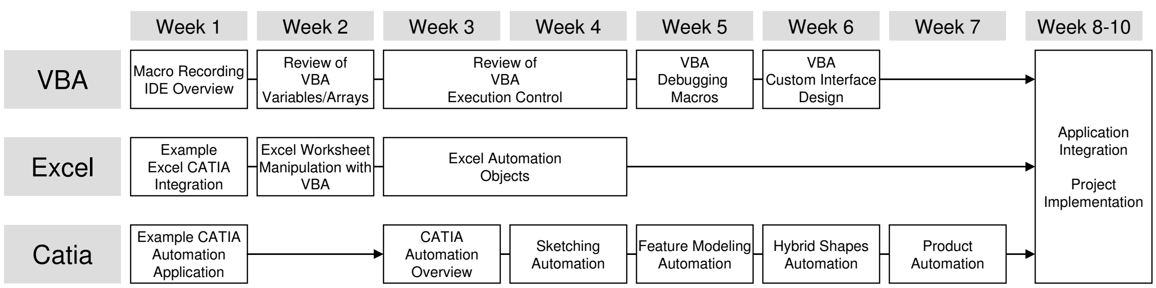


Figure 2. Course Topic Overview

Figure 2 shows a summary of the various topics covered over the course of the term. It can be seen that there are three topic streams that are covered concurrently. The first develops a student's background in VBA programming. For some students this is a review of their exposure to VBA taken in their required Computer-Science programming class. Given that students have several options to complete this requirement, those who have taken courses that use other languages (e.g. C++, Ada, Python) are learning VBA for the first time. Simultaneously they are being exposed to Excel macro programming through exercises that use VBA to retrieve and manipulate data stored in worksheets. Excel also provides a familiar environment for introducing students to Object-Oriented Programming. Students at best have a superficial understanding of this topic depending on their choice of programming course. This helps to ease them into the much more complicated object structure used in CATIA. As can be seen from the figure, the three streams converge towards the end of the term where integration between Excel, CATIA and interfaces developed in VBA is used to develop an automation application.

Teaching Strategy

Types of CAD Automation

To facilitate a systematic approach to teaching this subject several classes of CAD automation problems have been defined and are introduced to the students. These are as follows:

- *Part Configuration Automation:*
  This involves the automatic creation of a parametric model using user inputs through the VBA programming interface. The model can be built either from scratch or it can involve retrieving an existing model and changing valves of parameters to create a variant of the model. The latter approach to some extent duplicates capabilities that exist through the CAD system's GUI e.g. Design Tables. Automation of this type is particularly useful when the part geometry is controlled by physical equations.
- *Product Configuration Automation:*
  This class of automation involves the creation of different product structures by adding, removing or interchanging parts. This can also include changes to individual parts controlled by part configuration automation. As with the previous, this automation exceeds capabilities that are available using GUI functions such as Part Family Tables. Its power comes when logic must be applied in selecting components to build the configuration e.g. choosing and locating fixture elements on a fixture plate to locate and clamp a workpiece.
- *Data Retrieval:*
  This type of automation involves accessing part and product parameters, geometric and structural details (design history). One use of this type of automation is for data mining of CAD databases to match parts with characteristics similar to those defined in a search pattern.
- *Integration:*
  This can take on several forms the most common of which is the integration of a parametric modeling system with a database or spreadsheet. This provides a foundation for automating downstream activities such as machine design, physics-based modeling, Bill-of-Materials management and process planning.
- *Optimization:*
  This is the most complicated form of CAD automation where an environment is setup to vary a set of design parameters that drive changes to the geometry of a model while allowing one or more output parameter (e.g. weight) to be measured from the model. The output parameter is optimized by an engine that systematically changes the design parameters.

Students focus primarily on part/product configuration and integration automation in this course due to time limitations.

When to Use CAD Automation?

As part of exposure to these different types of CAD Automation classes students are given a series of questions that should be asked in deciding whether or not to invest the time and cost in developing an automation application. These questions are:

1. Are there large numbers of CAD users who need to reuse variants of the same components (parts or products) repetitively?

2. Does the time and cost expended in repetitive modeling warrant an automation solution?
3. Are there modeling tasks that would be practically impossible to do manually?
4. Does an automated solution enhance the productivity of a non-skilled user of the CAD system in question?
5. Can an automated solution enhance productivity of downstream product development activities?
6. Can automation facilitate integration of activities?
7. Can automation reduce the time for managing large numbers of CAD models?

Instruction in VBA using Excel

As can be seen from Figure 2, the first two weeks of the term are spent bringing students up to speed on VBA. To make this a more applied exercise rather than a study of programming syntax and a review of concepts and methods, Excel is used as the primary application at this point. Since students are familiar with how a spreadsheet works it is easier to use this understanding to show how manual interaction with a spreadsheet can be automated using a VBA macro. It is also easier to develop simple, practical exercises that retrieve and manipulate data from spreadsheet cells using a program. One example of such an exercise was a macro that ran a statistical analysis on machined part dimensions. Since students have performed similar exercises in their Quality Assurance class, this exercise helps to impress on them the benefit of automation in making analysis more efficient.

The VBA *Integrated Development Environment* (IDE) is also introduced at this time. Students become familiar with how to write and comment code, navigate through projects, create program structures (e.g. modules, forms, classes) and retrieve and manipulate properties of objects. They also learn how to use the *Macro Recorder* to create code snippets that can be integrated into their own macros so saving time in development.

Several texts were reviewed on Excel VBA programming[10,11,12,13]. It was felt that to support hands-on tutorial exercises, reference [10] was the best. This text has step by step instructions on how to create macros for manipulating data in Excel workbooks. Its weakness lies in explanations for programming syntax and concepts. Other texts while better in this regard tend to be too wordy and as such do not lend themselves to use in a tutorial setting. Having a tutorial type instructional resource with exercises that are design and manufacturing oriented would be ideal.

Instruction on CATIA Automation Objects

Students are provided with an example CATIA automation program at the start of the term primarily to motivate the explanation of automation and to show what they are expected to be able to do by the end of the term. Instruction on CATIA Automation starts in the third week following their exposure to VBA with Excel. This involves getting them acquainted with the object structure that captures a 3D parametric model in CATIA. This structure is quite expansive and must be selectively studied. Heavy use of the CATIA on-line documentation is necessary to do this as no instructional resource has been identified to teach this.

By correlating objects accessible through programming with geometry, sketches, features, bodies, parts and products, students get to see object-oriented programming being applied in a context that they are familiar with. For example a concept such as inheritance can be

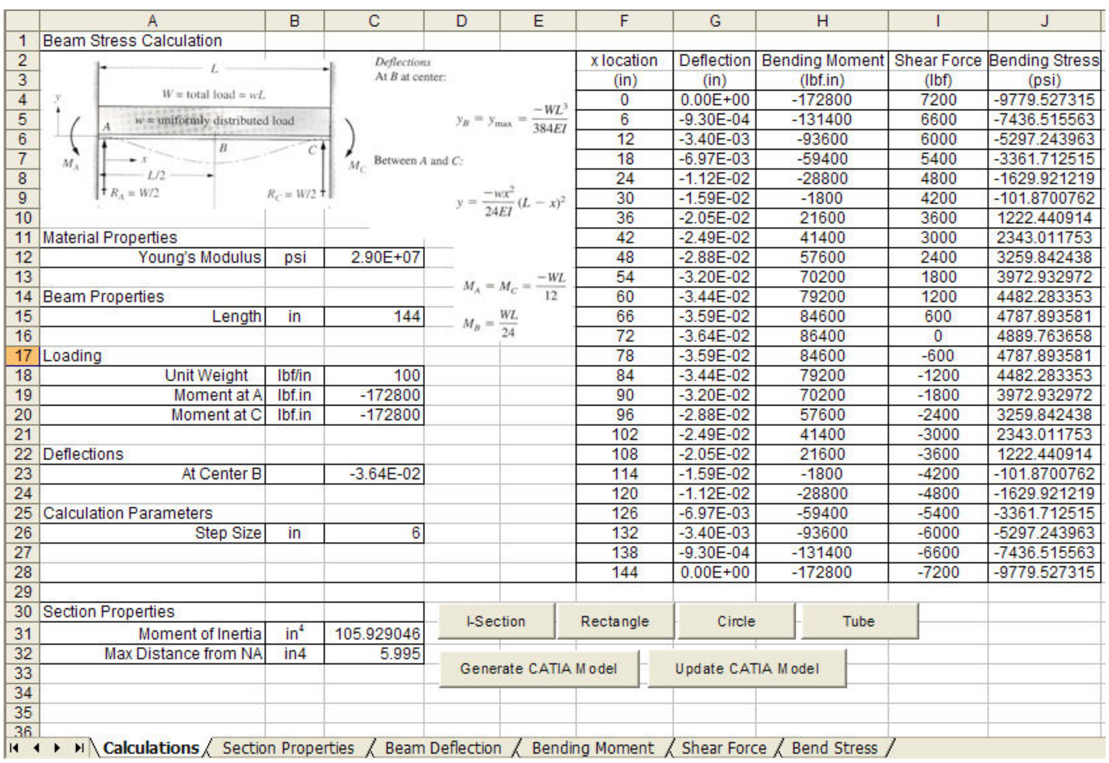
demonstrated using the *SketchBasedShape* class from which a *Prism* and finally a *Pad* or *Pocket* are derived. As shown in Figure 2, the study of CATIA automation starts with sketching automation and progresses through features, hybrid shapes (jargon for surface models) and products.
As with Excel heavy use is made of examples generated by the embedded macro recorder during tutorials to help in understanding how to programmatically access and manipulate the underlying object structure of a CATIA parametric model.

Assignments and Project Samples

<u>Automation of Beam Design and Modeling</u>

Homework assignments are designed to have students progressively apply VBA programming techniques using Excel and CATIA. Beam design automation is one example of an application that has been used for this purpose. It can be used to provide effective exposure to the *Part Configuration, Retrieval* and *Integration* types of automation described previously. The sequence of assignments and the objectives for each are as follows:

1. *Set up of Beam Deflection and Bending Stress Worksheet:* This exercise provides a review of Excel and how to set-up a spreadsheet to perform analysis of a beam that is rigidly supported at both ends supporting a uniformly distributed load. All equations are set-up as functions directly in the spreadsheet. Properties for only an I-Beam section are used.



Beam Stress Calculation

Material Properties		
Young's Modulus	psi	2.90E+07
Beam Properties		
Length	in	144
Loading		
Unit Weight	lbf/in	100
Moment at A	lbf.in	-172800
Moment at C	lbf.in	-172800
Deflections		
At Center B		-3.64E-02
Calculation Parameters		
Step Size	in	6
Section Properties		
Moment of Inertia	in^4	105.929046
Max Distance from NA	in4	5.995

Deflections
At B at center:

$$y_B = y_{max} = \frac{-WL^3}{384EI}$$

Between A and C:

$$y = \frac{-wx^2}{24EI}(L-x)^2$$

$$M_A = M_C = \frac{-WL}{12}$$

$$M_B = \frac{WL}{24}$$

x location (in)	Deflection (in)	Bending Moment (lbf.in)	Shear Force (lbf)	Bending Stress (psi)
0	0.00E+00	-172800	7200	-9779.527315
6	-9.30E-04	-131400	6600	-7436.515563
12	-3.40E-03	-93600	6000	-5297.243963
18	-6.97E-03	-59400	5400	-3361.712515
24	-1.12E-02	-28800	4800	-1629.921219
30	-1.59E-02	-1800	4200	-101.8700762
36	-2.05E-02	21600	3600	1222.440914
42	-2.49E-02	41400	3000	2343.011753
48	-2.88E-02	57600	2400	3259.842438
54	-3.20E-02	70200	1800	3972.932972
60	-3.44E-02	79200	1200	4482.283353
66	-3.59E-02	84600	600	4787.893581
72	-3.64E-02	86400	0	4889.763658
78	-3.59E-02	84600	-600	4787.893581
84	-3.44E-02	79200	-1200	4482.283353
90	-3.20E-02	70200	-1800	3972.932972
96	-2.88E-02	57600	-2400	3259.842438
102	-2.49E-02	41400	-3000	2343.011753
108	-2.05E-02	21600	-3600	1222.440914
114	-1.59E-02	-1800	-4200	-101.8700762
120	-1.12E-02	-28800	-4800	-1629.921219
126	-6.97E-03	-59400	-5400	-3361.712515
132	-3.40E-03	-93600	-6000	-5297.243963
138	-9.30E-04	-131400	-6600	-7436.515563
144	0.00E+00	-172800	-7200	-9779.527315

Figure 3. Main Calculation Sheet with Buttons for Running Macros for Different Beam Sections

2.

3. *Automation of Bending Equations and Standard Section Properties:* Using VBA, functions on the spreadsheet for calculating deflection and stresses are replaced by macros. Calculation of the section properties is also implemented in VBA and broadened to include other standard beam sections (rectangle, circular, tubular etc.). Macros are executed through buttons on the worksheet (see Figure 3). This provides students with their first programming exercise using VBA to retrieve, manipulate and post data on the spreadsheet.

4. *Creation of User Interfaces to Control Beam Design Automation:* Figure 4 shows examples of VBA *Userforms* created for streamlining data IO to Excel and CATIA. Students get practice creating their own forms and using these to control execution of their macros.

5. *Extracting Section Properties for Non-Standard Beam Sections:* Models of custom beams are created that conform to the assumptions used in bending theory. Macros are written to extract section properties from these models and to use these to calculate and plot deflection and stress over the beam length. In addition to demonstrating to students how the beam design spreadsheet can be used to calculate deflection and stresses for custom beam sections it also shows how mass properties can be extracted from a CAD model using automation.

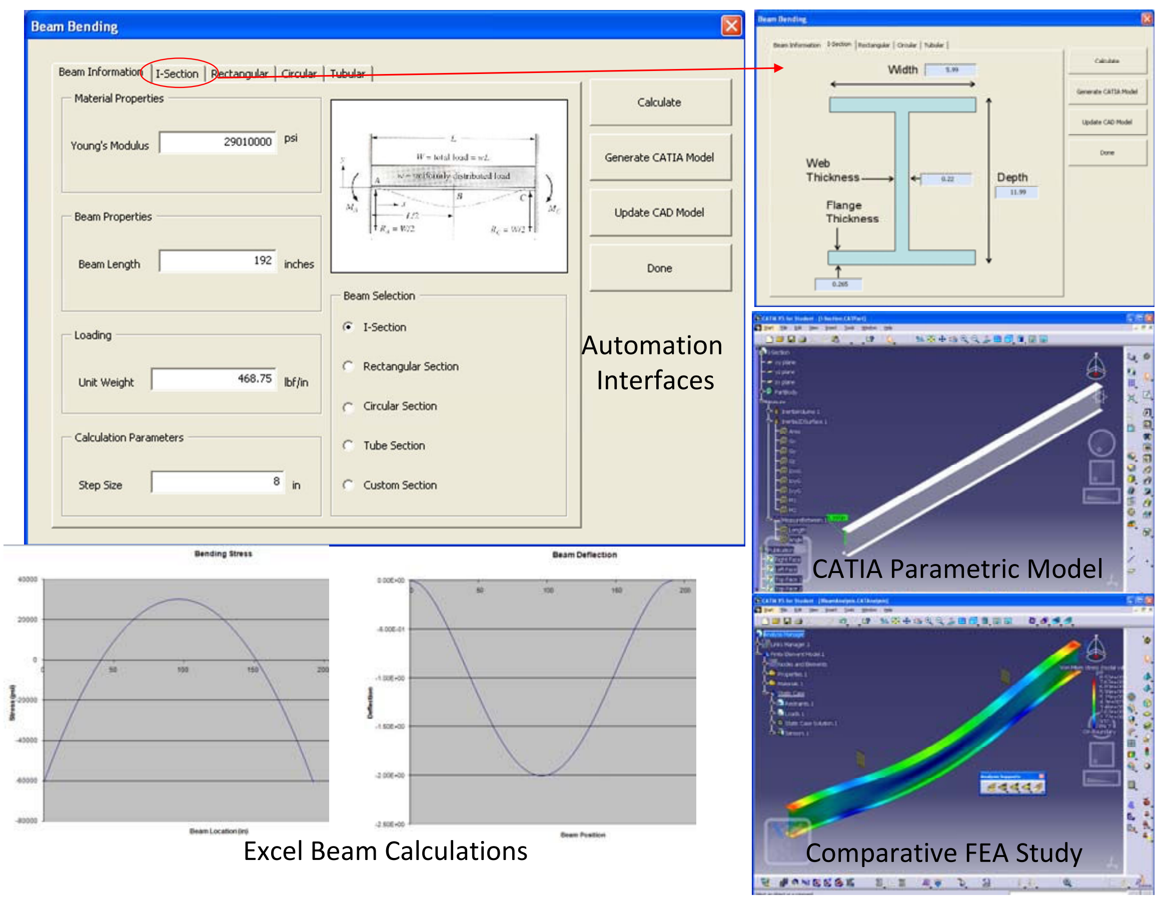


Figure 4. Beam Design Interfaces and Results

Product Configuration Automation

To provide exposure to using automation for configuring a product an assignment is given that requires the programming of VBA macros that switches components in an assembly. Figure 5 shows an example of this assignment using an idler pulley that is comprised of three major components (*pulley*, *frame* and *bracket*). A simple interface is created that allows a variant of each of these components to be numerically selected. By using *Publications* to simplify matching of geometry used in assembly constraints, one of three variants of these components can be used to create 27 different product configurations some of which are illustrated in the figure. Through this assignment students become acquainted with how to manipulate assembly models using VBA programming.

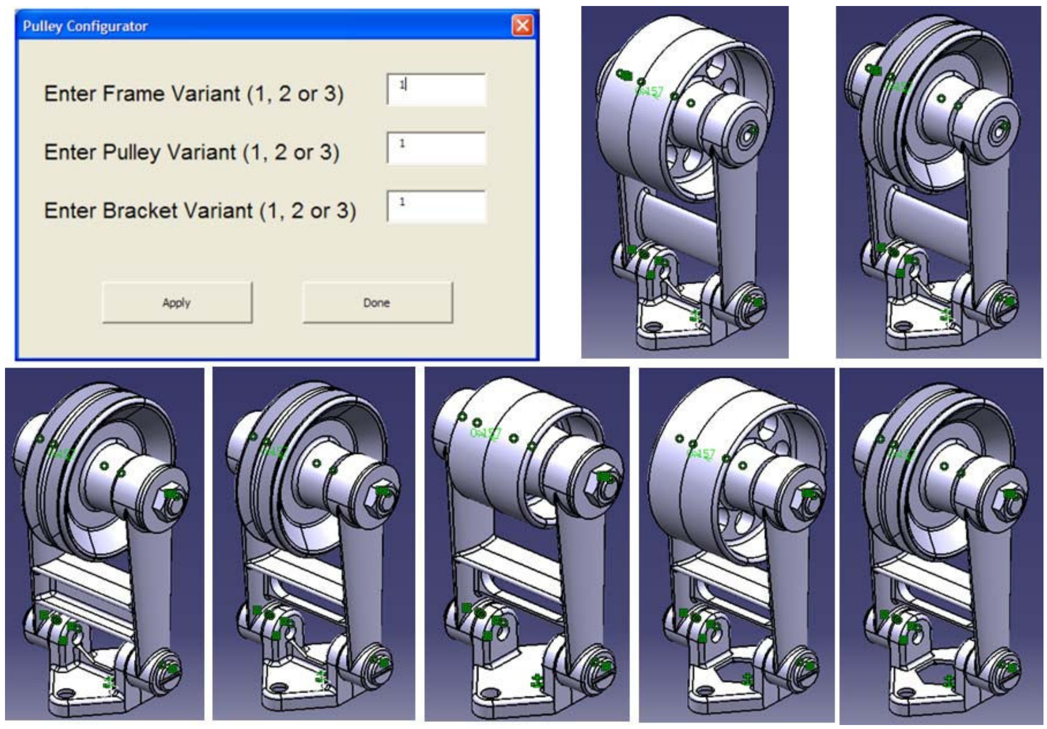


Figure 5. Examples of Automation Generated Product Configurations

Automation of Airfoil Design

The beam design assignments discussed previously provide a structured approach to learning how to develop CAD automation solutions. Students are required to work on a term project that gives a more open ended experience. One example of a project used in this course is the automation of the design of an airfoil. Each student has the opportunity to select the type of airfoil they wish to design. Examples include wings, hydrofoils, spoilers, propellers and wind turbines. They must conduct background research to determine how to model the base shape of the airfoil. As can be seen from Figure 6, this should allow them to model the leading and trailing edges of the airfoil and set up section planes where sketches for airfoil section geometry will be created (Steps 2 and 3). A construction line is created on each sketch connecting the leading and trailing edges to assist in extracting cord lengths for the wing (Step 4). Section geometry is based on the 4-digit NACA series[9], the parametric equations for which are given in Figure 7. These equations are set up on an Excel spreadsheet to create points on a section profile based on a NACA number (e.g. 4415) and a cord length. Since cord length varies along the airfoil, these must be retrieved from the CATIA model by measuring the length of the construction line created on each sketch (Step 5). Profile point coordinates are updated on the spreadsheet based on each cord length and passed back to CATIA to create splines on each sketch (Steps 7 and 8). Sketches are lofted (Multi-section solid or surface feature in CATIA) using the leading and trailing edges as guide curves to create the final model of the airfoil.

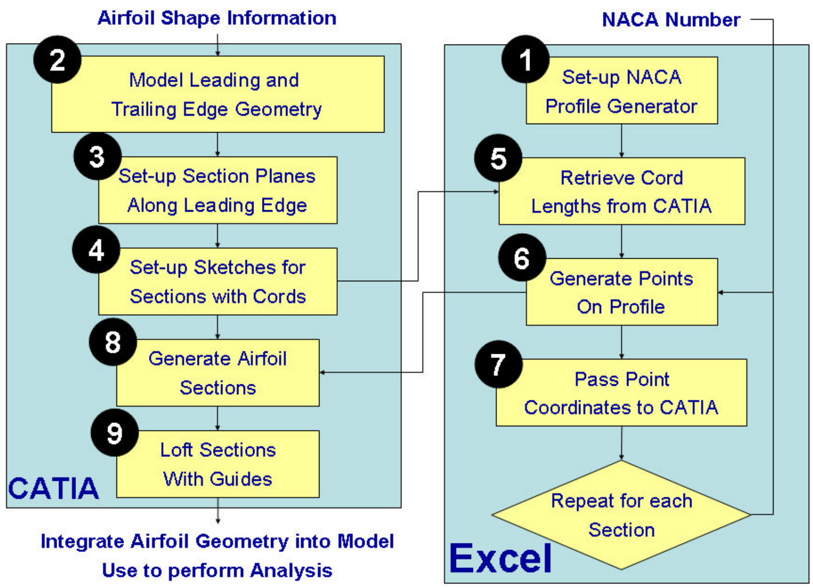


Figure 6. Steps Used in Airfoil Automation Design Project

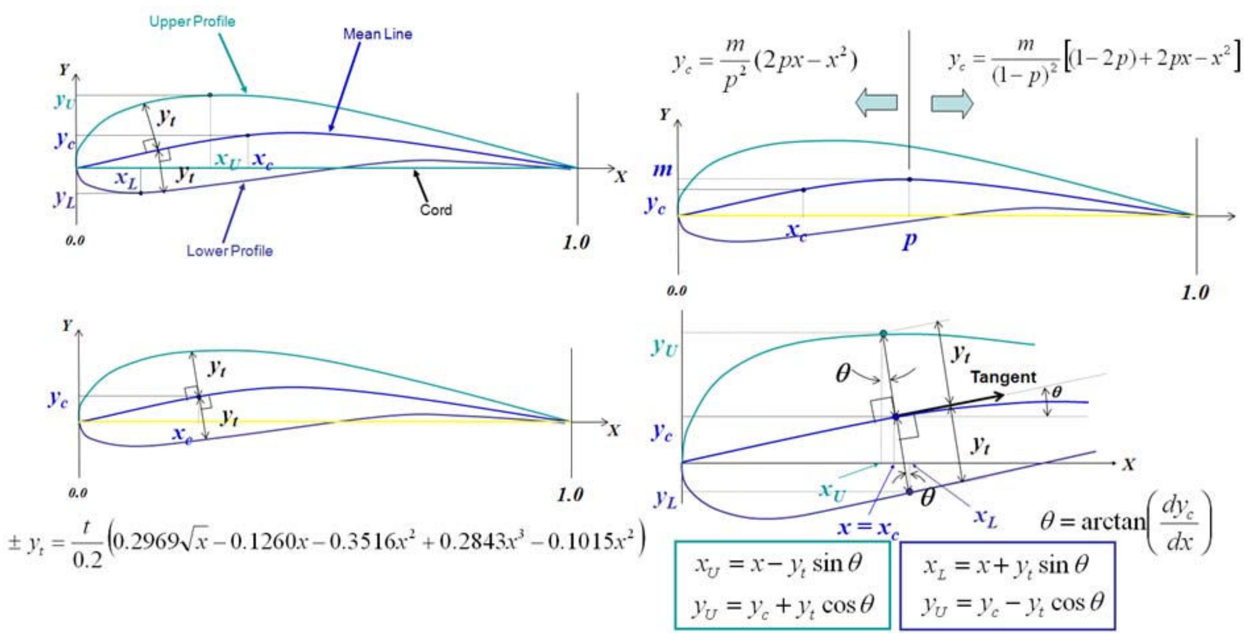


Figure 7. Parametric Equations for Generating NACA Section Profiles[9]

In addition to setting up the automation described in Figure 6 for their airfoil selection, students must also design and implement an interface to assist in modifying the airfoil geometry. An example of this interface is given in Figure 1. Both the shape of the wing and the section profile can be modified through this interface. An example of a project completed by a student is shown in Figure 8. This automates the design of a ship's propeller. In addition to controlling the blade section profiles and the outline of the blade the automation also allows the pitch of the blade to changed

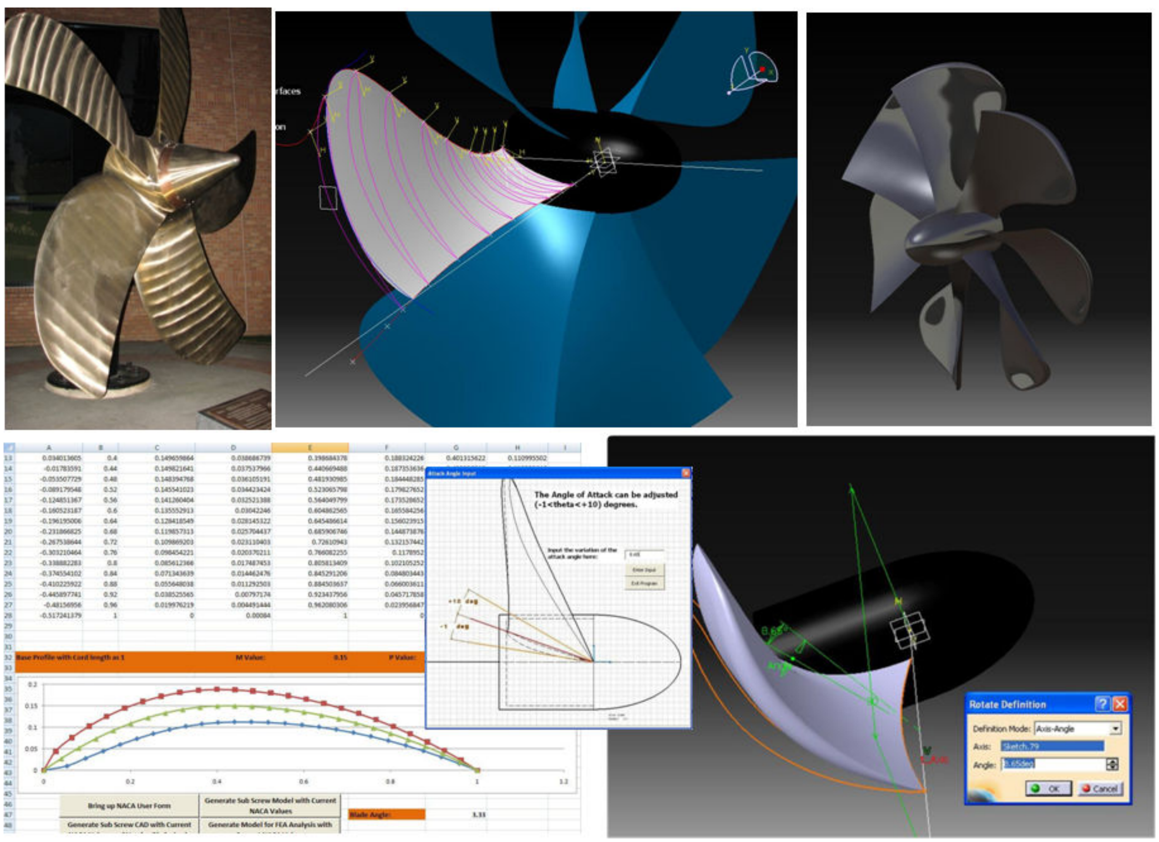


Figure 8. Example of Student Project for Automating Design of a Propeller

Challenges, Potential Improvements and Final Thoughts

For most IT-CAD/CAM students the experience obtained from this course is significantly different to what they have acquired in their other classes. This presents both challenges and rewards to teaching the subject. One of the greatest challenges is that programming does not come naturally to many of the students. Their prior experience in this area which comes primarily from within the context of a Computer Science course using a language such as C, C++ or Visual Basic is not always a solid foundation to build upon. This is because the focus there is on concepts and methodology and much less on application. For technologists who learn best through taking a hands-on approach to problem solving, this results in a disconnect between the instruction they receive in programming and their appreciation of its practical value for problem solving. To compound this, a crowded curriculum makes it difficult for these skills to be reinforced as they need to be to become a core part of a Technologist's skill set. Admittedly more can be done to reinforce the value of programming in different courses. Along these lines one improvement that can be made is to require CAD/CAM majors to take the Visual Basic programming option for their Computer Science course. A more drastic change would be to develop a computer science course specifically for technologists. This could incorporate a language like VBA, Excel and a math programming language such as Mathlab. A more application oriented approach would be taken to teaching this. While this has been discussed, it is difficult to implement due to resource constraints.

Another challenge comes from the constraint of having to teach the material in a 10 week term versus a 15 week term under the trimester system. Given that significant time must be spent in reviewing programming concepts, there is not enough time to delve into the CATIA Object Structure in as much detail as desired. Both the Excel and CATIA VBA *Macro Recorders* are extremely helpful in reducing the amount of time in creating examples to explain program development and also in helping the students create code segments with the functionality they need for their project. However, this can be a double edged sword when code generated by this tool is used without proper vetting and understanding of how it works.

Instructional materials also present challenges. As discussed earlier an appropriate text for use in a tutorial setting that mixes relevant programming exercises with explanations of programming concepts is difficult to find. A VBA Excel book with an engineering problem solving focus would be ideal. Materials for instruction using the CATIA Automation VBA interface are even less available. Both of these areas present opportunities for textbook development.

Overtime it is envisaged that a set of working automation examples will evolve from the projects such as the airfoil automation example presented in this paper. These can assist the instruction by providing modules that students can study to better understand how to efficiently develop their own macros. These can also be organized into a library that allows reuse to speed up program develop.

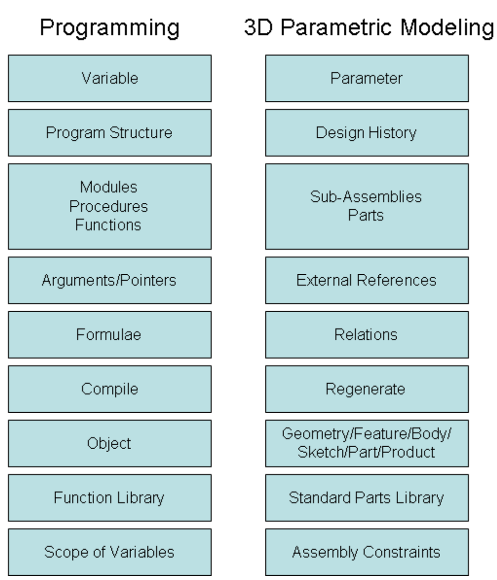


Figure 9. Comparison Between Programming and Parametric Modeling Concepts

Finally, it is important to note that there is a strong correlation between programming and 3D parametric modeling. Figure 9 shows how concepts in these areas relate. For example, variables in programming are conceptually similar to parameters that control size and shape in a CAD model and program structure which is procedural for VBA is likewise similar to the design history captured in a part or assembly. There is an opportunity in this course to synergistically reinforce concepts in these two domains. Some of this is currently happening as explained earlier with the correlation of CATIA Automation Objects and the modeling entities that are created during a typical modeling exercise with the GUI. More can be done in this regard.

Conclusions

This paper presents an overview of a new course in CAD Automation that utilizes VBA, Excel and CATIA. The instructional approach builds upon skills developed for VBA programming in Excel to encompass use of VBA in creating and manipulating parametric 3D CAD models in CATIA. Overviews of assignments and projects illustrate the progression in learning culminating in an automation project that integrates many of the techniques introduced over the term. In addition to introducing students to a new approach for CAD modeling, they get an opportunity to see programming work in solving real engineering design problems. By using Excel and VBA they also get to see that they have easy access to a programming environment which will increase the likelihood that they continue investing in developing this ability throughout their careers.

Bibliography

1. http://www.spatial.com/
2. Corney, J., Lim, T., *3D Modeling with ACIS*. Saxe-Coburg Publications, 2001.
3. http://www.plm.automation.siemens.com/en_us/products/open/parasolid/index.shtml?stc=usiia400109&gclid=CLKxsv62k58CFSgVagodb1L1_g
4. Lee, K. *Principles of CAD/CAM/CAE*. Addison Wesley, 1999.
5. Mortenson, M. *Geometric Modeling.* 2nd ed. Wiley, 1997.
6. Shah, J., Mantyla, M. *Parametric and Feature-Based CAD/CAM*. Wiley, 1995.
7. Spens, M., *Automating SolidWorks 2009 using MACROS.* Spiral bound version available at Amazon.com.
8. Sutphin, J., AutoCAD *2006 VBA: A Programmer's Reference. Springer-Verlag*, New York, 2005.
9. Abbott, I., Von Doenhoff, A., *Theory of Wing Sections.* Dover, 1959.
10. Simon, J., *Excel Programming*. 2nd ed. Wiley Publishing (Visual Series), 2005.
11. Birnbaum, D., Vine, M. *Microsoft Excel VBA Programming.* Thomson Course Technology, 2007.
12. Darlington, K., *VBA for Excel Made Simple*. Made Simple Books (an imprint of Elsevier), 2004.
13. Kelly, J., *Excel 2003 VBA Programming*. Wiley Publishing (Visual Series), 2005.