2006-1641: GUIDED CAE SOFTWARE LEARNING MODULES FOR THE UNDERGRADUATE MECHANICAL ENGINEERING CURRICULUM

Frank Fisher, Stevens Institute of Technology

Dr. Frank Fisher has been an Assistant Professor in the Department of Mechanical Engineering at Stevens Institute of Technology in Hoboken, NJ since August 2004. Dr. Fisher earned BS degrees in Mechanical Engineering and Applied Mathematics from the University of Pittsburgh, Masters degrees in Mechanical Engineering and Learning Sciences (School of Education and Social Policy), and a PhD in Mechanical Engineering, all from Northwestern University. Prior to his current appointment he spent for two years as a post-doctoral research associate for the Biologically Inspired Materials (BIMat) Center at Northwestern University.

Constantin Chassapis, Stevens Institute of Technology

Dr. Costas Chassapis is Department Director and Professor of the Department of Mechanical Engineering at Stevens Institute of Technology. He received BE, ME, and PhD degrees in Mechanical Engineering from The City University of New York.

Guided CAE Software Learning Modules for the Undergraduate Mechanical Engineering Curriculum

Abstract

Under development are a series of guided learning modules illustrating the use of CAE tools and their relationship to the engineering fundamentals covered within the core undergraduate curriculum. Learning supports embedded within these modules will explicitly reinforce basic engineering fundamentals and highlight how these principles relate to the types of engineering problems that are encountered in practice. Ultimately envisioned is a library of such modules which would enable the seamless incorporation of CAE tools throughout the undergraduate curriculum. These modules leverage two important characteristics of these CAE tools: 1) the visualization capabilities of the CAE software, and 2) the ability to solve more complex problems than those typically covered within a traditional lecture-based format. Such tools will provide additional curricular tools which professors may use to adapt more inductive and active teaching methods within their classes.

Introduction

While advances in computer technology, and in particular Computer Aided Engineering (CAE), have drastically changed the practice of professional engineering, adaptation of such technologies within the undergraduate educational system has typically been less disruptive. The anticipation of a fuller implementation of such technologies into undergraduate engineering education provides the opportunity to critically re-assess the content of the undergraduate curriculum, the manner in which this material is taught, and the potential for these technologies to enhance the undergraduate experience for our students.

A common dilemma when considering the undergraduate curriculum is the proper balance between "teaching fundamental theory" versus "teaching applied software". While the establishment of a sound base of engineering fundamentals within our students is perhaps the primary goal of the undergraduate curriculum, increasingly there is a legitimate incentive to expose students to the proper use of different engineering software tools in preparation for their professional careers. While efforts to include such computational techniques have included the development of an elective upper-level undergraduate course[1], the practical difficulty of implementing these changes within and throughout the curriculum, and in particular the difficulty of integrating software into an already crowded syllabus, has been noted.[2] In addition, Petroski eloquently points out the danger of introducing students to the use of CAE tools without proper training, noting that "the computer is both blessing and curse for it makes possible calculations once beyond the reach of human endurance while at the same time also making them virtually beyond the hope of human verification".[3]

The current manner in which our department addresses the "theory versus software" debate is to incorporate CAE tools within a required junior-level Modeling and Simulation class, where students are introduced to various commercially-available CAE software packages and their use as a problem-solving tool in engineering. While this class has been successful in introducing our undergraduates to a number of software packages such as SolidWorks, CosmosWorks, ANSYS, ProEngineer, and Simulink, we have become increasingly dissatisfied with the artificial (and perhaps arbitrary) "theory versus software" divide that such a class implicitly presents to the students.

This dissatisfaction led us to consider alternative methods in which to cover the use of these software tools within the curriculum. After considerable thought, it became clear that, despite the practical difficulties associated with its implementation, the proper place to present such CAE tools is alongside the traditional engineering fundamentals covered within the curriculum. Covering such software tools in this manner has a number of potential learning benefits. In particular, such an exposure can help students put into context the types of idealized problems typically covered early in the curriculum, and demonstrate for students the relevance of these illustrative, and often simplified, examples to the types of complex problems which they will encounter in real-world engineering practice. Others have also recognized the importance of exposing students to more complex systems and problems earlier within the curriculum.[4]

This motivation has led to an ongoing program to develop a series of guided learning modules to incorporate standard CAE software tools within and throughout the individual classes comprising the core undergraduate mechanical engineering curriculum. The goal of this program is to enable the seamless incorporation of CAE tools to complement the fundamental principles covered in the undergraduate curriculum. Ultimately envisioned is a library of such modules which would be suitable for deployment throughout the undergraduate curriculum.

The design of these modules will leverage two important characteristics of CAE: 1) the visualization capabilities of CAE software, and 2) the ability to consider more complex problems than those typically discussed within a traditional lecture/homework format. In addition, learning supports within these modules will *explicitly* reinforce basic engineering fundamentals covered within a particular class and highlight how these principles relate to the types of engineering problems that are encountered in practice. Such tools will provide professors with additional curricular tools to migrate their classroom discussion from an authoritarian model of instruction to a more student-centered approach with the goal of developing higher-level thinking and engineering skills within the undergraduate curriculum.

Incorporation of the modules within the Mechanical Engineering curriculum

A distinguishing characteristic of the curriculum at Stevens Institute of Technology is its extensive "core curriculum," a tradition since its founding in 1871. At the heart of the core curriculum is an eight-semester design sequence known as the "Design Spine." As illustrated in Table 1, the five components of the Mechanical Engineering curriculum are the Engineering Core (including the Design Spine), Integrated Science Core, Mechanical Engineering Core and Electives, and a Humanities and Social Sciences Sequence. The first five design courses (Terms I-V) expose students to design issues associated with each of the main engineering disciplines, while the last three design courses (Terms VI-VIII) are domain-specific and constitute the capstone design sequence. The Design Spine provides integration of design with the science and engineering-science courses, in many cases with courses taken concurrently, to provide context and reinforcement across classes. To fulfill the competencies required of engineering graduates, the design spine promotes an increased emphasis on topics relating to professional practice, communication skills, teaming, project management and economics of design, skills that are developed progressively and reinforced throughout.

Term I	Term II	Term III	Term IV	Term V	Term VI	Term VII	Term VIII
Ch107 General Chemistry I	Ch116 General Chemistry II	Ma221 Mathematical Analysis III	Ma227 Mathematical Analysis IV		ME322 Engineering Design VI (IPD)	ME423 Engineering Design VII	ME424 Engineering Design VIII
Ch117 General Chemistry Laboratory I	Ch118 General Chemistry Laboratory II	PEP201 Physics III	PEP202 Physics IV	E321 Engineering Design V			
Ma115 Mathematical Analysis I	Ma116 Mathematical Analysis II		E232 Engineering Design IV	E344 Materials Processing	E355 Engineering Economy	E421 Engineering Economy & Design	
PEP101 Physics I	PEP102 Physics II	E231 Engineering Design III	E246 Electronics & Instrumentation	E243 Probability & Statistics	ME354 Heat Transfer	ME483 Control Systems	ME491 Mfg. Processes & Systems
CS115 Introduction to Computer Science	E122 Engineering Design II	E245 Circuits & Systems	ME225 Dynamics	ME358 Machine Dynamics & Mechanisms	ME345 Modeling & Simulation	Mech. Engineering Elective	Mech. Engineering Elective
E121 Engineering Design I	E126 Mechanics of Solids	E234 Thermodynamics & Energy Conversion	ME335 Thermal Engineering	ME342 Fluid Mechanics	ME361 Design of Machine Components		
E120 Eng. Graphics I						Elective	Elective
E101 Seminar							
Hu Humanities	Hu Humanities	Hu Humanities	Hu Humanities	Hu Humanities	Hu Humanities	Hu Humanities	Hu Humanities
PE115 Physical Education	PE116 Physical Education	PE225 Physical Education	PE226 Physical Education	PE325 Physical Education	PE336 Physical Education		

Engineering Core including Design Spine; Science Core; Mechanical Engineering Requirements; Integrated Humanities and Social Sciences sequence; "Free" Electives; Mechanical Engineering Electives

Table 1. The Mechanical Engineering undergraduate curriculum at Stevens Institute of Technology.

While the Design Spine has been successful in closely integrating aspects of engineering design to the core fundamentals of the engineering curriculum, a similar integration of CAE software across and thorough the curriculum has to date been lacking. Currently, in-depth coverage of the use of CAE software tools in the context of engineering problem-solving is for the most part postponed until Term VI, where it is the primary focus of the Modeling and Simulation (ME345) course. The reasoning behind such placement within the curriculum is that students first must have a thorough grounding in (mechanical) engineering fundamentals before such software can be adequately presented to the students. In addition, coverage during Term VI prepares the students to use such tools on subsequent internship/co-op assignments and the capstone Senior Design project, as well as provide the relevant modeling skills desired by industry upon graduation.

As envisioned, the proposed library of CAE learning modules will change our current coverage of CAE software by enabling such tools to be incorporated across and throughout the curriculum as shown in Table 2. Such an arrangement will provide illustration of the use of CAE tools and their relationship to the engineering principles covered within each of the classes comprising core mechanical engineering curriculum. Taking advantage of the visualization and analysis capabilities of the CAE tools will provide instructors the opportunity to implement inductive teaching techniques within their classes (i.e., having students solve complex problems as part of a pre-class assignment, followed by discussion of the problem within the traditional lecture format). In addition, learning supports and additional information embedded within the modules will explicitly reinforce the core engineering principles covered within the class, provide review of previous material, foreshadow more advanced material covered later in the curriculum, and indicate how this materials relates to the "big picture" of engineering in practice.

Course	Cosmos Works	Cosmos Flow	Cosmos Motion	Ansys	Flotran	Pro/E	Pro/ Mechanica	SolidWorks	Pro/ Manufact.
Engineering Graphics								red	
Mechanics of Solids	red			yellow				red	
Dynamics			red					red	
Fluid Mechanics		red		yellow	yellow			red	
Machine Design/Mechanisms			red						
Modeling and Simulation	red	red	red	yellow	yellow	blue	blue	red	
Heat Transfer	red	red		yellow	yellow	blue	blue	red	
Design of Machine Comps.	red	red	red	yellow		blue	blue	red	
Manufacturing Processes						blue	blue		blue
Intro CAD						blue	blue		blue
Design for Manufacturability						blue	blue		blue
Advanced Heat Transfer	red	red		yellow	yellow	blue	blue	red	
Simulation and Modeling	red	red	red	yellow	yellow	blue	blue	red	
Advanced Manufacturing						blue	blue		blue
Adv. Mechanics of Solids	red			yellow		blue	blue		
Adv. Structural design	red			yellow		blue	blue		
Finite Element	red	red	red	yellow	yellow	blue	blue	red	

red = SolidWorks; yellow = Ansys; blue = Pro/E

Table 2. Course map describing incorporation of engineering software tools into various classes within the Mechanical Engineering curriculum.

In addition to their incorporation within the undergraduate curriculum, it is anticipated that an available library of such modules will benefit other department constituencies as well as highlighted in Table 3. As discussed later in this paper, what separates these modules from other commonly available CAE software tutorials (often provided by the various software companies) is the incorporation of additional information within the module which *explicitly* relates the analysis covered within the module to the fundamentals of the undergraduate curriculum. This additional information embedded within the module, supplemental to the "point-and-click" nature of a standard software tutorial, is a critical distinguishing characteristic of the design.

Pedagogy of the proposed module design

As summarized by Felder[5], a multi-level model of college learning was developed by Magolda to track the epistemological development of college students.[6] Briefly, the model contrasts the lowest form of learning ("Absolute Knowledge") with the highest form of knowledge ("Contextual Knowledge"). Students in the former category believe that:

> **Absolute Knowledge (negative connotation):** "Lecture is the only legitimate form of instruction, and a good teacher is one who provides clear and unambiguous statements of the things the students are expected to know and gives repeated practice in the required problem-solving procedures. *The student's task is to memorize the knowledge, practice the procedures, and repeat both on examinations.[6]*

This is in contrast with students at displaying the highest qualities of learning, "Contextual Knowledge":

> **Contextual Knowledge (positive connotation):** Contextual knowers… question the assumptions underlying all assertions, their tolerance of ambiguity (which deters them from rushing to accept the first plausible explanation that arises), their inclination to use both logic and intuition in their investigations, and their unwillingness to transfer judgments made in one context to another context without critical evaluation, *could almost stand as a definition of what first-rate scientists and engineers do.*[6]

The quality of learning has also been characterized by the *approach* that students take to learning:[7]

> **Surface Approach (negative connotation):** Students who take a surface approach memorize facts but do not try to fit them into a coherent body of knowledge and follow routine solution procedures without trying to understand their origins and limitations. To them, studying means scouring their texts for worked-out examples that look almost identical to the homework problems so they can simply copy the solutions.
>
> **Deep Approach (positive connotation):** In contrast, students who take a deep approach try not just to learn facts but to understand what they mean and how they are related to one another and to the students' experience. They have… a tendency to question conclusions offered in lectures and readings. They cast a critical eye on each statement or formula or analytical procedure they encounter in class or in the text to see if it makes sense to them.

UNDERGRADUATE STUDENTS - (*) Currently, students intermittently use different software packages but the skill-set quickly erodes if consistent usage is not promoted. - (*) Engineering software competency represents an attractive skill-set for students entering the workforce. - (*) Such modules promote lifelong learning by providing students self-directed learning experiences within the undergraduate curriculum. - (*) Undergraduates would benefit from seeing throughout the curriculum how such tools are used in practice. - Motivated students can delve deeper into CAE packages only briefly covered in the curriculum.
GRADUATE STUDENTS - Introducing graduate students to different software packages provides an opportunity to further their skills and competencies as they conduct their research. - Graduate students may be asked to TA classes using software tools with which they are not familiar; these modules will enable them to function more effectively for the benefit of both the TA and the class.
FACULTY - (*) Introducing CAE tools into "theory-heavy" classes can be cumbersome and time-intensive. Properly developed, these modules will enable faculty to efficiently incorporate these tools into such classes. - For classes where significant time is invested on a particular CAE package(s), well-constructed modules will allow faculty to "offload" software-specific questions, freeing more time for higher-level discussion. - As engineering software becomes increasingly advanced and complex, it is often difficult for faculty to remain current as to the latest software packages that are available. - More efficient updating/replacing of the core engineering software and related materials.
ALUMNI AND INDUSTRIAL PARTNERS - (*) Alumni may need to learn new (or newer versions) of these software packages, thus these modules will provide a valuable service to our alumni. - (*) As part of professional training programs, industry may have occasion to train engineers/technicians on the use of current software packages.

Table 3. Beneficiaries of the proposed modules. Items marked with (*) are considered particularly significant.

It is perhaps not surprising that not all undergraduates reach the higher levels of learning desired of a successful undergraduate education, when the stereotypical "chalk-and-talk" methods of lecturing typically reinforce a "*Surface Approach*" to "*Absolute Knowing*". The challenge for engineering educators is to develop innovative methods to incorporate learning opportunities that lead students to develop these deeper approaches to learning and these higher forms of knowledge, yet do so in a manner that does not significantly compromise the rate or amount of material that can be covered.

In addition, Felder and co-workers have written at length about the different learning styles of undergraduate engineering students.[8,9] For example, student learning styles have been characterized according to the following five dimensions: sensory vs. intuitive, visual vs. verbal, inductive vs. deductive, active vs. reflect, and sequential vs. global.[10] While student learning styles can be expected to vary within a given class, the average college instructor is likely to be described as an intuitive, verbal, deductive, reflective, and sequential learner.[11] This is reflected in the stereotypical "chalk-and-talk" teaching style prevalent in engineering education.

However, a number of studies have shown the benefits of inductive teaching methods (such as problem-based and project-based learning) and active learning (engaging students in activities other than listening to lectures); see Reference [8] and references therein. While inductive teaching methods have been proposed and used at the undergraduate level, the focus of such efforts are typically larger-scale, hands-on projects that may require significant faculty and student effort and time investment. Thus the goal of the current effort is to develop learning modules based on the use of CAE software tools to enable professors to efficiently adopt more inductive-based instructional methods within the classroom.

Description of a prototype learning module

An example of the design of a module for use in E126 (a hybrid statics-strength of materials class) is based on the analysis of a car-jack using CosmosWorks as shown in Figure 1. The analysis starts with a SolidWorks solid model of the car jack, which can either be provided as part of the module or included as a separate file provided as part of the assignment. (Because SolidWorks is covered earlier within the curriculum, such that students could have previously created this module as part of an Engineering Graphics assignment.) One benefit of the proposed modular design is that the modules can be revisited throughout the curriculum; for example, the car-jack module can be viewed from both a kinematic as well as structural problem. Using the approach proposed here, the module guides students through the specific steps necessary to use the software (a standard "point-and-click" tutorial), while at the same time *explicitly* tying such an analysis into the principles currently being covered within the class, such as free-body diagrams (see Figure 2).

Properly written, the "point-and-click" portion of the module will be complete and contain all necessary steps for students to quickly assemble and complete the analysis, enabling professors to use such modules as the basis for out-of-class assignments without having to devote valuable class time to covering the specifics of the use of the CAE software package. However, while standard tutorials are sufficient for leading students through the necessary steps of using these software tools, the overall goal of the current project is much more extensive. Specifically, we want to use the introduction of engineering software tools within the undergraduate curriculum as a means to promote inductive and student-centered learning opportunities. This would be done by supplementing the "point-and-click" instructions of a standard software tutorial with additional content information that can be discussed in relation to the analysis being presented in the tutorial. Examples of such information could be both fundamentals-related as well as modeling- and analysis-related. To prevent overloading the student with too much information, the material will be presented in a hypertext (webpage) format, allowing students to control the amount of “supplementary content material” that they wish to access. An additional benefit of a web-based module approach is that these modules can be continuously updated and supplemented with additional content material via, for example, linking with new material to reflect updated versions of the CAE software. Multiple modules will also be able to access the same supplemental material (consider, for example, multiple structural analysis models linking to a common Hooke’s Law page). As appropriate, an evaluation component at the end of each module will prompt students to answer questions related to the analysis, such as discussion of the relationship between the software solution and the engineering fundamentals being covered within the class, and interpretation of the results of their model, including assumptions implicitly involved within the modeling procedure.

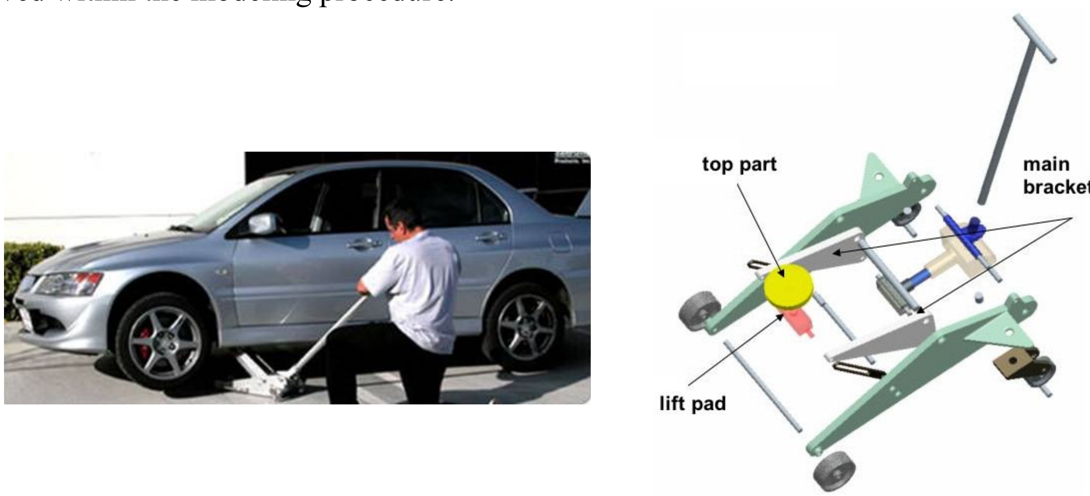

Figure 1. Example of car-jack problem for hybrid statics-strength of materials class module.

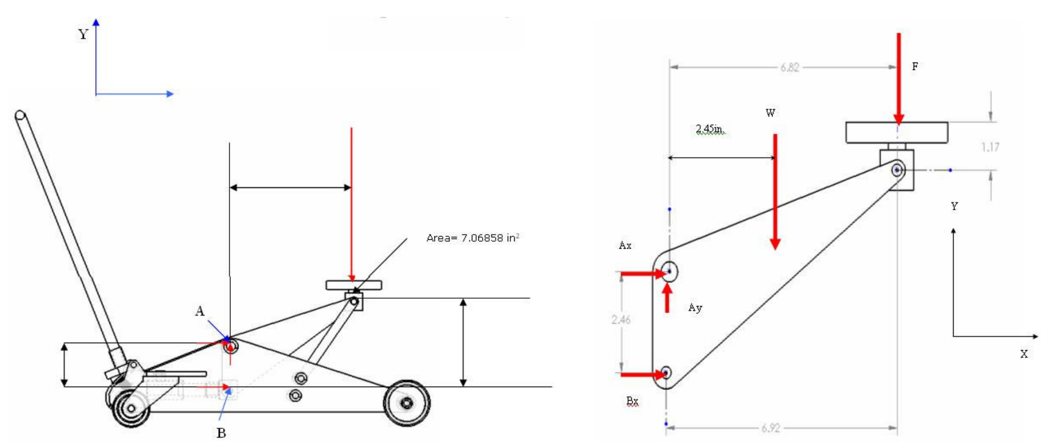

Figure 2. Free-body diagram of the car-jack main bracket.

Given the solid model and proper guidance, the model can easily be adapted by the student to investigate the response of the car-jack to any number of changes to the system (see Table 4).

For example, it is straightforward to alter the materials of the jack or the load of the car, while more complicated changes such as the dimensions of the car-jack elements and the height of the car could be changed with minimal effort. In addition, such modules highlight for the students more advanced engineering topics. Examples of such "advanced" engineering topics include the finite element method (see Figure 3), stress concentrations, failure analysis, and manufacturing and design issues. The goal of briefly touching on such topics within the module is *not* an in-depth discussion of these principles, but rather to demonstrate how the engineering principles covered within the class contribute to the "bigger picture" their engineering studies. This is the critical distinction between "standard tutorials" and the self-guided learning modules described within. By only covering the step-by-step processes necessary to use the software, standard tutorials implicitly reinforce a surface approach to Absolute Knowledge. By comparison, the goal of the current effort is to use the engineering software as a vehicle to facilitate within students a deeper approach to Contextual Knowledge.

1	Would the jack function properly if the car was replaced by a Ford F150?
2	How does changing the material of the lift pad (to steel/aluminum/polymer/composite/etc) effect the jack performance?
3	How does reducing the thickness of the main bracket by 10% effect the design?
4	How does reducing the main bracket shaft diameter by 10% effect the design?
5	Discuss differences in performance using Bracket B (supplied) in the design.

Table 4. Potential modifications of the car-jack module for in-module student assignments.

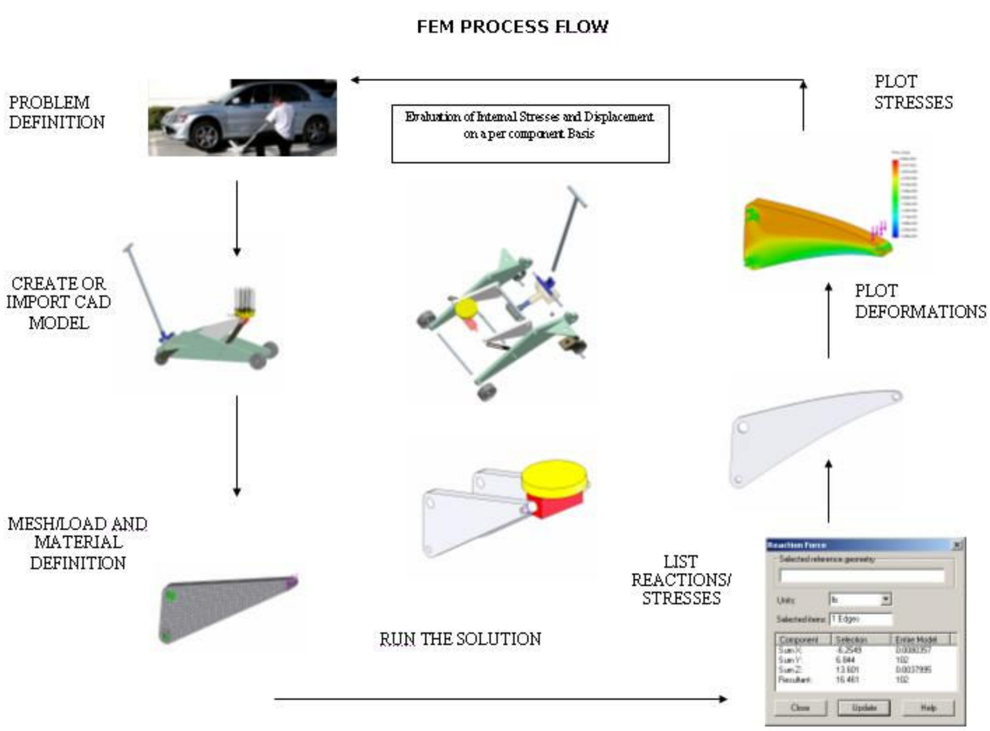

Figure 3. Illustration of the process flow for a finite element method based on the car jack example.

Evaluation

A critical aspect of the module development will be appropriate user-testing, of both the modules themselves as well as their implementation within the curriculum, to ensure that the learning objectives behind the module design are being met. A standard novice-expert test protocol will be implemented to ensure that the students are able to use the modules to complete the modeling assignment and that the content material embedded within the module is at an appropriate level. We note that one benefit of the proposed web-based module format design is that these modules be easily modified, updated, and extended based on student feedback; particularly with respect to additional questions or comments that students may submit through a feedback mechanism that will be standard in the module design.

The second component of the evaluation will be the implementation of the modules throughout the mechanical engineering curriculum on a class-by-class basis. Particular concerns in this respect are the ability of the instructor to implement these modules within the syllabus, and whether such modules are effective in enabling professors to adopt more student-centered, inductive approaches within the classroom. We will also assess the time commitment associated with the use of these modules, from the perspectives of both the students and the instructors. We envision that workshops or instructional materials may be necessary to encourage faculty buy-in with respect to the module use. Focus groups of students using the modules within the undergraduate curriculum, as well as structured interviews with faculty participants, will provide input and feedback into the module development program. Of particular interest will be whether the instructors felt that the modules were successful in fostering higher-level thinking and problem-solving characteristics of the students in the content domain covered within their particular class.

Summary and Future Work

We are in the initial stages of developing a series of guided learning modules to facilitate deployment of CAE software simulation tools within the undergraduate mechanical engineering curriculum. Primary goals of this module development effort include:

1. to demonstrate the use of CAE tools and their relationship to the engineering principles covered within the core curriculum;

2. to enable the seamless incorporation of these CAE tools throughout the curriculum;

3. to leverage the visualization capabilities of the CAE software to facilitate the adaptation of more complex and ill-defined problems within the traditional lecture format;

4. to provide a mechanism to reinforce previously covered material, as well as foreshadow advanced material to be covered later in the curriculum, to enable students to see the “big picture” of the undergraduate curriculum;

5. to equip professors with additional curricular tools to adapt more inductive and active teaching methods within traditional lecture-based classes.

Use of these guided modules will allow students to effectively use the CAE software to quickly model/simulate a subset of engineering problems related to a core class in the undergraduate curriculum. While the modules will lead the students through the "point-and-click" steps necessary to use the CAE software for a given problem, additional learning supports within the modules will present and reinforce the basic engineering and modeling principles covered within the modules, while explicitly relating these concepts back to the core mechanical engineering curriculum. Such self-contained modules will enable the class discussion/lecture to focus on engineering and modeling fundamentals rather than software-related issues and questions. It is anticipated that the continued development and implementation of these software learning modules will lead to the incorporation of small, subject-matter specific CAE modules in classes that focus on theory, as well as provide faculty the opportunity to effectively and efficiently incorporate software-intensive simulation components into their existing classes.

Early versions of this format have been successfully deployed in a graduate-level off-campus class with positive student feedback, and we are currently developing modules for implementation at the undergraduate level such as the one described above. Once user-testing of these initial modules is complete, their incorporation within the curriculum will be assessed. Evaluation of the modules within the mechanical engineering curriculum, faculty buy-in and acceptance, and the impact of the modules on student learning will be a critical component of the project and will be reported at a later date.

Acknowledgements

The authors acknowledgement the support of the Research & Innovation in Engineering Education program led by the Center for Innovation in Engineering & Science Education (CIESE) at Stevens.

Bibliography

1. Baker, J.R. An Elective Course on Computational Techniques in Structural Mechanics with a Balance in Theory and Application. Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition. 2005. Portland, OR.
2. Mazzei, A. An Approach for in-Class Learning in Mechanical Engineering Design Subjects. Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition. 2005. Portland, OR.
3. Petroski, H., To Engineer Is Human: The Role of Failure in Successful Design. 1992, New York: Random House.
4. Craig, N., V. Addison, M. Maher, and W. Peters. Integrating Complex Systems Study into the Freshmen Mechanical Engineering Experience. Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition. 2005. Portland, OR.
5. Felder, R.M. and R. Brent, The Intellectual Development of Science and Engineering Students. Part 1. Models and Challenges. *Journal of Engineering Education*, **93**(4): p. 269-277, 2004.
6. Magolda, M.B., Knowing and Reasoning in College. 1992, San Francisco, CA: Jossey-Bass.

7. Felder, R.M. and R. Brent, The Intellectual Development of Science and Engineering Students. 2. Teaching to Promote Growth. *Journal of Engineering Education*, **93**(4): p. 279-291, 2004.
8. Felder, R.M. and R. Brent, Understanding Student Differences. *Journal of Engineering Education*, **94**(1): p. 57-72, 2005.
9. Felder, R.M. and L.K. Silverman, Learning and Teaching Styles in Engineering Education. *Engineering Education*, **77**(5): p. 647-681, 1988.
10. Felder, R.M., Reaching the Second Tier: Learning and Teaching Styles in College Education. *Journal of College Science Teaching*, **23**(5): p. 286-290, 1993.
11. Center for Teaching and Learning, Teaching for Inclusion. http://ctl.unc.edu/tfi1.html. Last accessed: March 6, 2006.