



Preparing Students for Industry by Integrating Commercial Software into Coursework

Prof. Joseph A. Untener, University of Dayton

Joe is a professor of Engineering Technology at the University of Dayton. With degrees from General Motors Institute and Purdue University, and experience with General Motors and other engineering and manufacturing companies, he teaches courses in Mechanical Engineering Technology, and co-authors text with Robert L. Mott.

Prof. Robert L. Mott, University of Dayton

Robert L. Mott, P.E. is professor emeritus of engineering technology at the University of Dayton. He is a member of ASEE, SME, and ASME. He is a Fellow of ASEE and a recipient of the ASEE James H. McGraw Award. He is a recipient of the SME Education Award. He holds the Bachelor of Mechanical Engineering degree from General Motors Institute (Now Kettering University) and the Master of Science in Mechanical Engineering from Purdue University. He has authored the textbooks; Applied Fluid Mechanics 7th ed. and Machine Elements in Mechanical Design 5th ed., published by Pearson/Prentice-Hall; Applied Strength of Materials 5th ed. with CRC Press and the 6th edition is in preparation.

Mr. Buck Jones, Engineered Software

Jones graduated from the University of Tennessee with a Bachelor's of Science in Engineering Physics and Christian Brothers University with a MBA specializing in Finance and Marketing. He has spent over 25 years in engineering, sales, marketing, and training positions. His experiences and education have afforded him the opportunity to develop collaborative agreements with customers, marketing teams, manufacturers, engineers and vendors which enhanced product value; have created prototypes, created product opportunities, and value analysis and value engineering.

**Preparing Students for Industry by
Integrating Commercial Software into Coursework**

Preparing Students for Industry by Integrating Commercial Software into Coursework

Abstract

The “automation” of engineering tasks has been an issue in engineering education since at least the time of the first handheld calculator. Whether or not to require the use of “trig tables” is an argument settled long ago, but the role of software in engineering education continues to be a topic of discussion.

This paper presents one model that addresses this topic from the development of the textbook all the way through the delivery of the student experience by individual faculty members. In this case, text book authors partnered successfully with an engineering software company and continue to work with instructors of the course to close the loop on the educational experience. The authors worked to integrate software seamlessly into a traditional text that has been successful over six previous editions. The result was a simple integration with no additional costs or contracts. Since downloads of the software are tracked by the software company, use of the software can be monitored, support can be provided, and the feedback loop can be closed to improve not only the learning experience, but also the textbook and the software.

The intention of this paper is to share one model for updating a text and a course to bring more practical application without losing the foundational work that supports it. This approach can result in a tremendous enhancement to a text, without requiring that instructors make a dramatic shift in their syllabi or teaching approach. It also yields the opportunity to close the loop with instructors who have adopted the text by providing feedback on the acceptance of the software as a viable component of the course.

Learning Objectives

Graduates are best positioned to serve industry well if their courses offer them *both* the undergirding foundational principles *and* an ability to apply those principles to current challenges. Industry expects to hire graduates who not only command the engineering principles, but also have some experience with application. In terms of pedagogical theory then, our students need undergraduate experiences that move them throughout Bloom’s taxonomy. While goals are often set to achieve the higher orders in the hierarchy, those are not achieved without the most basic foundation first, that of *knowledge*¹.

A tremendous amount of work has been conducted and documented relative to Bloom’s taxonomy applied to engineering technology education. It is well documented that limiting a course to lecture limits the depth of learning. Perhaps the most common element to research in this area is the concept of varying pedagogical methods, even within a given course. It is clear that there is great benefit to having at least a few different facets to the learning experience such

as lecture, independent reading, computer simulation, group discussion, and open-ended projects, to name a few.² One reason is that individual students have different learning styles and this multi-faceted approach will reach a broader base of students. Another though, is that different approaches to teaching move students to different levels within the taxonomy. A group discussion is not an effective way to convey basic *knowledge* of a subject, just as lecture is not a very good way to drive *synthesis*.

Much has been written about the use of computers as one such element of an enriched learning environment. Like most teaching methodologies, the introduction of a software component can be done well and it can be done poorly. “When technology is used for technology’s sake, it can obscure rather than illuminate.”³ Software components in engineering courses however, can also elevate the level of learning and allow students to move above the basic knowledge level and advance in both their capability and their confidence level. The use of application-oriented software, after gaining knowledge, can provide a platform that facilitates true exploration and innovation. Relative to perceived workload, students often view time spent in this activity differently than they do their time of “study” or “doing problems.” The automation of calculations provides for many more iterations, so failed design attempts are short lived as the students progress to successful conclusions with relative ease.

The objectives in our courses then should be to ensure that the information transfer takes place in an effective and efficient manner, provide other experiences that will match better with some learning styles, and also likely achieve different elements of the taxonomy.

Role of the textbook and traditional classroom environment

The role of the textbook in today’s courses, and really the entire publishing industry, is in a state of flux. One model on the continuum is one in which every student purchases a hardcopy textbook specified by the professor and that textbook, along with lectures three days a week, provide “the material” for the course. On the other end, courses are open sourced, free flowing experiences directed by an instructor with general course objectives in mind. The reality in most courses today is somewhere between those extremes, and it is difficult to know where the academy, and the publishing and learning software companies, will be moving over the next several years.

The transfer of knowledge and learning of the basic concepts is still critical to a solid engineering education. While there are many modes for that transfer of knowledge and building of basic skills, students reading, hearing, and seeing explanations, remains a generally effective method. A textbook, whether paper or electronic, with the fundamental principles and theory explained and illustrated, with some sample applications of those principles applied, remains a critical element in this level of student learning. Textbooks need to facilitate varying forms of pedagogy such as having a computer element to them, not just in terms of learning mode, but of the actual course content. Enriching that total course experience, bringing it to life, making it active,

introducing projects, and finding appropriate computer applications, becomes the role that most faculty take as instructors of the course.

It is clear that the knowledge transfer stage, that of truly developing a command of the principles, cannot be skipped, or the results can be disastrous. Great harm can result from polished looking results of a FEA or CFD that was generated by person who lacks a command of the engineering principles that drive the software. Traditional textbooks in their traditional format are, in fact, very good for gaining the knowledge that is the basis of learning the principles of engineering. Computer programs are no more a replacement for learning engineering principles than Wikipedia is a replacement for libraries. The role of the textbook, whether electronic or paper or interactive or mobile, remains an important source of student learning.

Use of commercial application software to enhance learning – one model

There are many ways for faculty to enrich the learning environment, vary pedagogical methodology, and move the level of student learning up Bloom's taxonomy, given that the students have already achieved the *knowledge* level. In engineering technology courses, one excellent method is the use of commercially available software, typical of the work environment encountered immediately after graduation.

This paper does not address the large all-encompassing software packages that all engineers should know. Nor does it address courses that are essentially built around a given computer application and training in the use of that software. Rather, this paper advocates for the more niche-oriented, but commercially available software as one relatively small element in courses in which the engineering fundamentals are still being taught.

The model presented here is for a sophomore level fluid mechanics course, and the textbook is *Applied Fluid Mechanics* by Robert L. Mott which was first published in 1973 and is one of the most popular fluid mechanics texts in Engineering Technology today. In its first edition, the role of the computer was not issue. As the book moved through six editions, the text moved with the times, introducing computing and then progressing and changing including computer programming assignments, enclosed discs, spreadsheet files, and specific "internet resources" listed at the end of the chapter⁴. The newest edition, the 7th, now co-authored with Joseph A. Untener, utilizes an integrated approach with a modeling system called PIPE-FLO[®] by Engineered Software Inc. in Lacey, Washington. PIPE-FLO[®] was not written for an academic audience. It is used throughout many industries to model fluid pipe and pumping systems.

The calculations required to correctly model fluid systems can be very involved and even tedious. The text relies heavily on the data and information from "*Technical Paper No. 410 – Flow of Fluids Through valves, Fittings, and Pipe (TP410)*", by the Crane Company⁵. This approach prepares students for real industrial positions, and exposes them to the extensive line of valves, fittings, and other products that are commercially available to solve real-world problems. This approach was taken from the very first edition of the text, with the objective of presenting an applied approach to teaching fluid mechanics that is based on solid fundamental principles

while using industry-standard data and methods of analysis that would prepare its users to become productive employees in industry very rapidly.

A major emphasis in this book is the analysis and design of pumped fluid flow systems and other pressurized or gravity fed systems. While including a wide variety of applications, many are of the type that those using this book will encounter while working in product-producing industries that employ industrial scale piping, valves, fittings, pumps, and controls. Other supportive chapters cover fluid properties, pressure measurement, and flow measurement.

Crane's TP 410 provides the bulk of the data and information about valves and fittings in the chapter on minor losses that adds to the prior development of energy losses due to friction in circular pipes and tubing and in noncircular sections. The combination of these principles is then employed in later chapters that develop the user's ability to analyze comprehensive fluid flow systems with or without a pump. Specifically, the book includes information about and flow resistance data for eight types of valves and eight styles of fittings, a discussion of the equivalent length technique for determining resistance factors, and data for friction factors in the zone of complete turbulence that are necessary to complete the calculations.

The book also includes a strong emphasis on the use of computer assisted methods for helping students to perform analyses of complex systems rapidly once they have mastered the problem solving procedures using manual techniques. Spreadsheets are introduced and students are encouraged to design their own or to enhance those included in the book. The goal is to encourage students to seek ways to become more efficient in their work while maintaining a strong emphasis on understanding the bases of the analysis methodology. This approach is then extended further by introducing students to the very comprehensive pipe-line system analysis software, PIPE-FLO[®], developed by Engineered Software, Inc. Through this multi-phased approach, students develop their knowledge and skills by steadily increasing the complexity of the systems with which they can work effectively. To accomplish these objectives, the consistent use of industry standard energy loss methodology and data are essential. Using Crane's TP-410 provides that consistency.

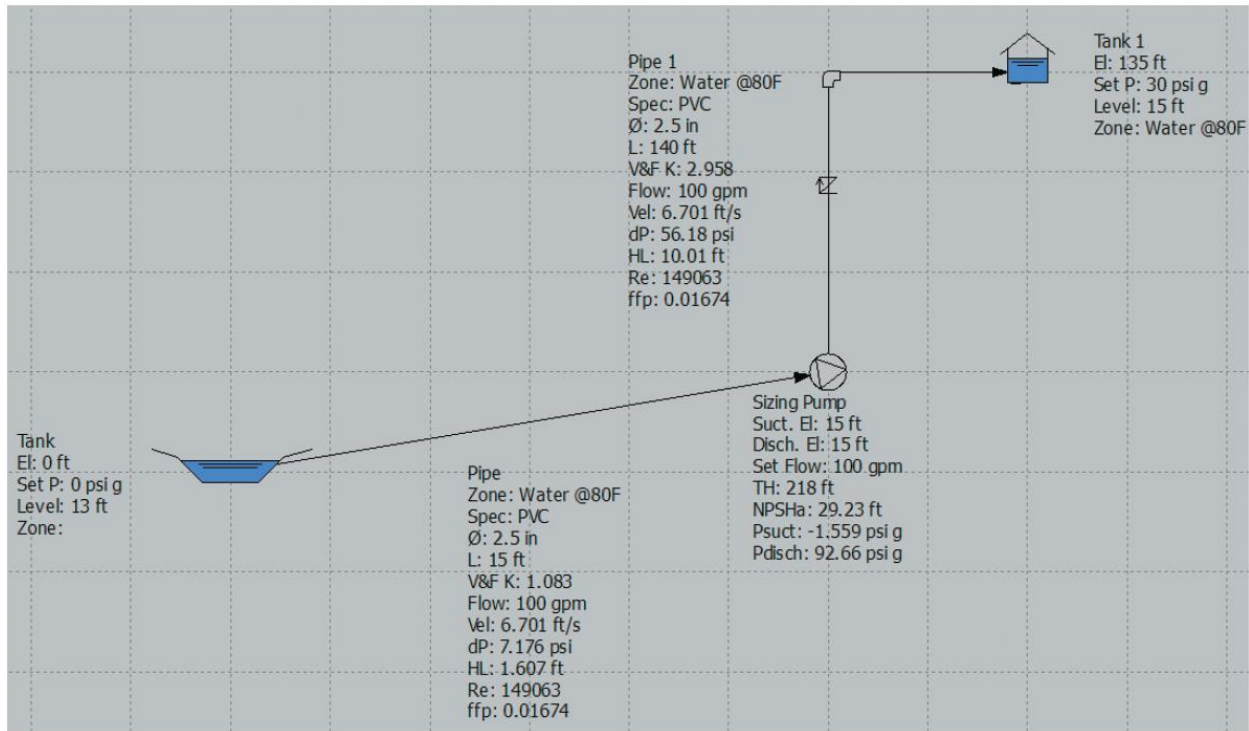
Methodology for integrating the software

The text, both in the preface and again when the first PIPE-FLO[®] problem appears, directs the reader to a website (<http://www.eng-software.com/afm/>) where ESI has provided a student-friendly landing page and shows a copy of their text to confirm the partnership. Students are directed through a simple download of the demo version of the software, the same demo that is actually available to anyone on the internet.

The first time the software appears in the book is in Chapter 8, once students have sufficiently learned the required background knowledge. The first use is a student's introduction to this element of the course, and since this element is critical to a different type of learning, this should be a friendly and positive introduction. Clearly we need to take measures to drive the initial student reaction to be something like, "Cool, this really makes this much easier and gives me tons more options." If instead, the initial student reaction is "This class was already hard, now I have to learn this computer stuff on top of all of that, and it so complicated," then that first impression will be very difficult to overcome.

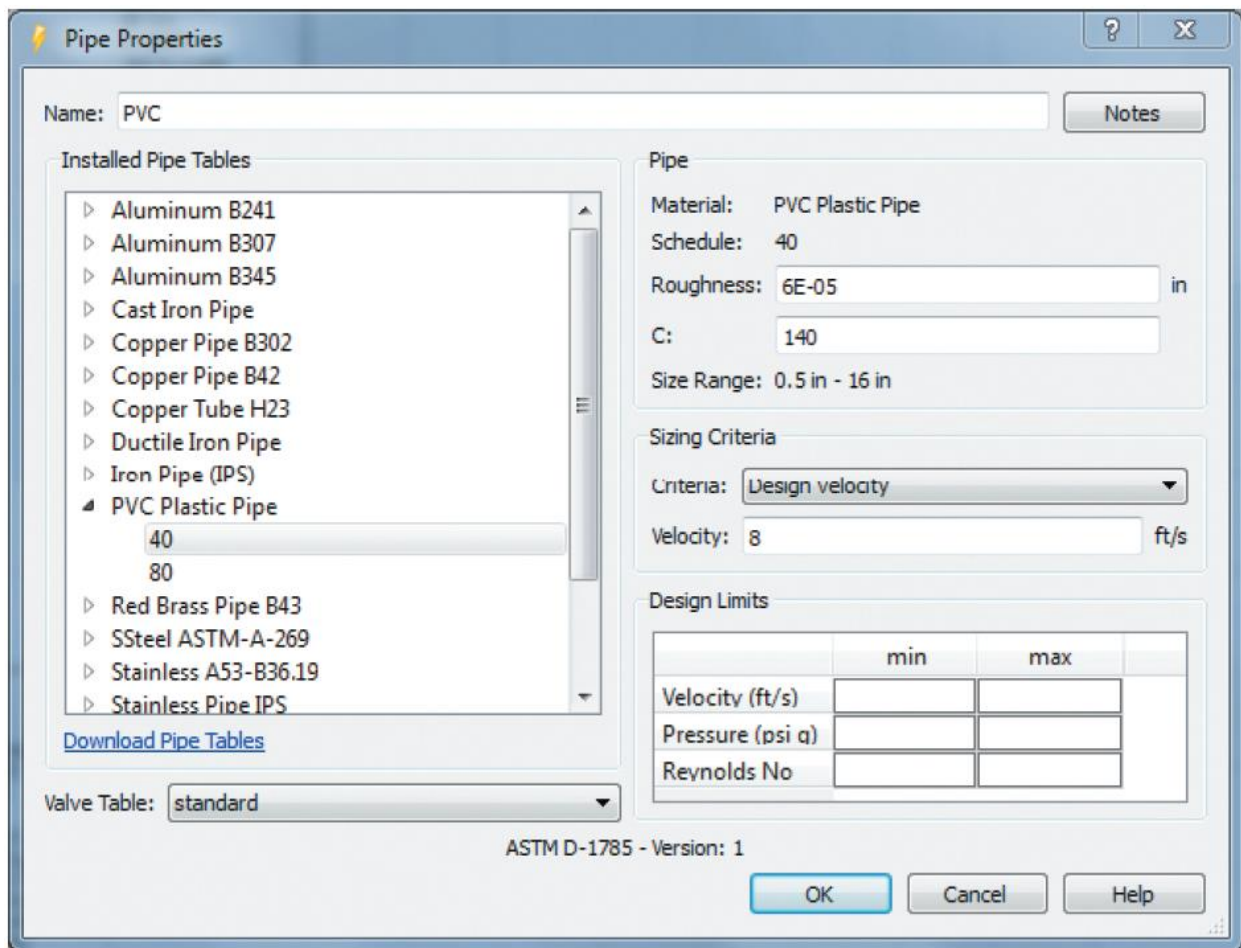
The first problem in the text is rather simple, but appropriate for this point in the course. In the following five chapters, as just a small part of each, the use of the software is increased, introducing new capabilities of the software, paralleling the increased complexity of the principles and problems in the text. A screenshot of one of the problems is shown below in Figure 1 below.

Figure 1 Simple fluid system modeled in PIPE-FLO®



Students at this point in the course are becoming very skilled at seeking out the many elements of required data from the text in both chapters and in the appendix. This requires substantial paging through to gather the required elements. In the software, one of the first things students notice is that the data is a bit more accessible. Not only do some basic clicks specify the pipe and the fluid, but all the properties associated with those are automatically populated. Students understand the “table lookup” that generated those values and of course know how to check any of that data as well. Another element that many students notice is more impressive: the sheer number of options available. They come to understand that while the book contains a substantial list of pipes and fluids and conditions, for example, there are many more available via a software program. The selection and the ease of specifying any pressure and temperature is empowering at this point in the semester. Figure 2 shows a simple data input page of the software, many fields of which are automatically populated, once the student has sufficiently constrained the necessary parameters.

Figure 2 Screenshot of typical data input

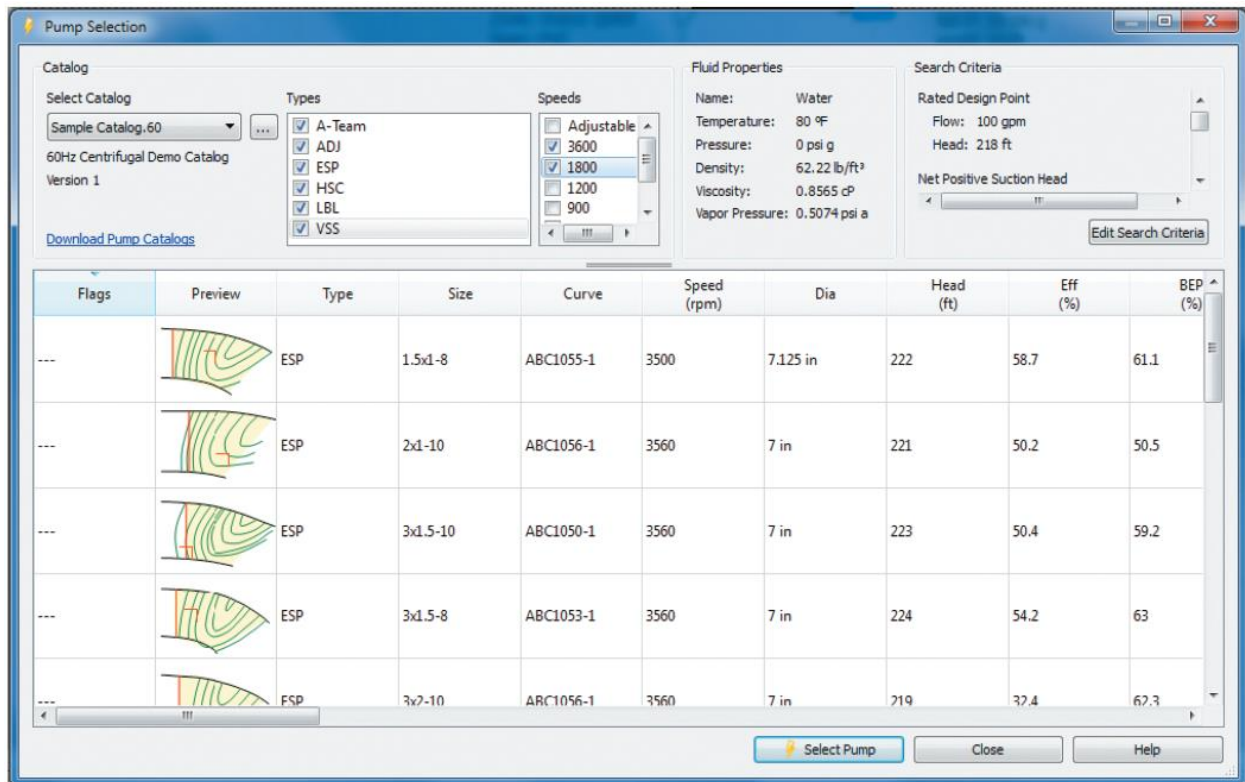


The generation of calculated results is seamless. Students at this point have manually manipulated many calculations for the course, as that is quite possible. It is important to know that this software is not performing CFD where there really is no manual alternative. In this model, students know how to do the calculations manually and will continue to do so throughout the course. With PIPE-FLO[®] problems though, once the known data are entered and the problem is sufficiently constrained, the values of interest are shown on the screen near the component to which they refer. Note in Figure 1 that all key output values for the pump, as specified by the user, are displayed next to that component.

Also with regard to results, student can find real value in the software and also get a sense that this exercise is truly a precursor to their employment experiences. Figure 3 shows commercially available pumps that meet the needs defined by the student. This single screen can only display

a small portion of the multitude of alternatives that the student can scroll through and select from. This is a tremendous advantage in that it goes well beyond the limited numbers of pumps listed in a traditional textbook, and also shows the students that they will literally be buying that pump, complete with model number and can even shop for that component just as they will after graduation. Additional learning can take place as the actual pump curves and operating points are accessible and correspond exactly to what the students have studied the entire term, but this time in a dynamic and interactive way.

Figure 3 List of potential pumps, along with curves, for application to the system



Key characteristics of this particular model

The model presented in this paper is only a slight variation of what is accomplished with some regularity throughout engineering technology education. These particular characteristics, however, are worthy of note as they are critical to the success of this particular model.

1. The software is not separate from the textbook. There is full integration of the commercial software into the text *and* a presence for the text on the commercial software site.

2. The software is a free, downloadable, demo version. This characteristic reduces barriers to adoption significantly. There is no cost issue or challenging technology/license issue and the software is actually the same software that is available to anyone via the internet.
3. The software used in the text is a respected commercial program that was not written for academic use. It is not a pervasive tool that is a part of all engineering education. This aspect is clear to students and they know that it is highly specialized, allowing working engineers to automate the process of doing the very calculations the students are doing in the course.
4. The course can run without this aspect. An instructor who, for example, has taught the course many times with this text in the past can simply ignore this facet of the text, and is no worse off than in previous terms. There are some pages, of course, devoted to PIPE-FLO explanation and problems, but they represent a small percentage of the text and they stand alone in a way that they can be omitted. The decision to add this facet to the course can be made next time, and is not dramatic or intimidating.
5. The software is reasonably easy for a beginning user to set up basic models. It does not require substantial narrative to get to an introductory level of skill. General explanations, guidance on the use of the software, some sample screenshots, and example problems efficiently achieve that goal.

Advantages of this model

This particular approach yields many opportunities relative to its alternatives:

1. This approach allows an introductory course to remain focused on developing principles, but still vary the pedagogy and move to higher level cognitive development. Faculty do not have to choose between a strictly traditional course format *or* training on the use of software.
2. Curricula are already overtaxed. As faculty we can always find more things that should be added to the curriculum, but we can rarely agree on removing anything. To teach the use of software we should not have to overhaul or add courses that are dedicated to training in the use of a particular software package. Rather, in existing courses, we should find ways to sprinkle in a bit of this element.
3. Expanding beyond the covers of the book allows for substantial input data, and perhaps more importantly, actual commercially available products, a hallmark of engineering technology education, to be used as design solutions.
4. Students complete a course with experience using commercial software that automates the processes learned in the course. It is done with professional and reliable software, but one that still requires a full understanding and does not cover for a lack of critical thinking or an understanding of principles.

5. Since the downloads are monitored by the software company, adoption of this can be tracked. The software company typically reaches out to those who download the demo version, and this close contact offers faculty an avenue for technical support.

This edition of the text has been available for only a year, so results are limited at this point. Feedback has been very positive though, and indicates that this aspect of the text has been a great addition and instructors indicate that they will continue to utilize it. One professor added this to the syllabus and took a “portfolio” approach, bundling these assignments together and evaluating various aspects such as problem statement, theory, hand calculations, and discussion as a part of the portfolio submission. Additionally, some students have reported that they like this aspect of the course because they see how this directly prepares them for their engineering careers.

Requirements and Limitations

This model offers many advantages and there is opportunity to leverage this approach to other textbooks and courses, but of course there are limitations as well.

1. This model requires a true partnership between textbook authors and a recognized commercial software company. Ensuring that individual interests are well-served by the arrangement is critical. Stakeholders include students, faculty, authors, publishers, and industry, and all must be remembered when establishing the partnership.
2. Matching print material with more dynamic software can be a challenge. A strong and stable software company must be part of the equation to ensure sustainability. Also the ability to adjust to software revisions between textbook updates must be planned.
3. As is always the case with experiential learning, the addition of this facet of the course does require faculty time and attention, and also time and attention of the class itself.
4. It is difficult to predict acceptance and the amount of adoption of this portion of the text, and hence the amount of textbook resource that can be dedicated to it.
5. The matter of using computer modeling is not fully settled. While most work concludes that higher learning takes place, there are indications in other cases that students sometimes don't fully engage in thinking in such exercises, but rather just enter data in an iterative fashion. More work needs to be done in this area and faculty need to be aware of this potential problem.

Conclusions

Many software products are such an integral part of a working engineer's life that no institution is going to graduate a student without experience. Every single mechanical program is going to include coursework in the 3D modeling, for example, just as circuit analysis software is a part of all electrical programs. All programs will also teach general tools such as mathematics

packages, spreadsheets and other packages. Industry hires graduates and rightfully expects that they have had training and experience with these large packages. Further though, many graduates enter a department where more “niche” software might be present. These packages do not have teams of hundreds of dedicated developers writing code and interacting with users. The packages are effective and reliable and require an understanding of the calculations. These types of software packages are bit more specialty in nature, and a very real part of our graduates’ working environments.

Software in its purest state is really just a way to automate what can be done manually. If a firm is working constantly with similar Bills of Materials for a particular family of products, they may have it automated. If a company is always working with fire suppression, or motor windings, or compressor performance, they likely have some automated tools to increase the efficiency of their constrained engineering staff. Our graduates will be prepared for this if we have introduced them to similar applications earlier. In addition, this element in a course reaches students with different learning styles and can drive higher level learning. The use of this software does not eliminate the need for engineers as engineering professionals need to run the software.

There are many similar approaches in use today and this details just another variation that can be considered for other courses and texts. As the role of the textbook and the publishing industry in general continue to seek out the new realities, models like this should be considered and further evaluated. The direction of textbook publishing and of educational software should be driven by faculty in the engineering technology discipline, with input from students, always considering the appropriate use of technology with student learning and preparation for industry as the focus.

References

1. Bloom, B. S. et al, “Taxonomy for educational objectives: Handbook I: Cognitive domains.”, New York: David McKay, 1956.
2. Bielefeldt, A.R. Pedagogies to Achieve Sustainability Learning Outcomes in Civil and Environmental Engineering Students. *Sustainability* 2013, 5, 4479-4501.
3. Cordell, R. New Technologies to Get Your Students Engaged. *Chronicle of Higher Education* 2011, 57, 36, B8-B10.
4. Mott R.L, Untener J.A, *Applied Fluid Mechanics*, 7th edition, Pearson, 2014.
5. *Flow of Fluids through Valves, Fittings, and Pipe*, Technical Paper 410, Crane Company, 2009.

6. Taraban, R., Anderson, E. E., DeFinis, A., Brown, A. G., Weigold, A. and Sharma, M.P. (2007), First Steps in Understanding Engineering Students' Growth of Conceptual and Procedural Knowledge in an Interactive Learning Context. *Journal of Engineering Education*, 96: 57–68. doi: 10.1002/j.2168-9830.2007.tb00915.x
7. Nirmalakhandan, N., Ricketts C., McShannon J., and Barrett S., Teaching Tools to Promote Active Learning: Case Study. *Journal of Professional Issues in Engineering Education and Practice*, Vol. 133, No. 1, January 2007, pp. 31-37.
8. Turner, M., Foreman C., and Sundararajan R., Computer Simulation Tools to Enhance Undergraduate Power Systems Education, ASEE National Conference Proceedings, June 2014.
9. Durfee, J. K., Richter D. C., Weiser, M., and Saad, H.S., Using Course Projects to Infuse Innovation Throughout the Undergraduate Experience in the Engineering and Engineering Technology Curriculum, ASEE National Conference Proceedings, June 2014.