

## **2006-377: MOTIVATING STUDENTS IN AN INTRODUCTION TO COMPUTING COURSE BY REQUIRING ANIMATED SOLUTIONS**

### **David Myszka, University of Dayton**

Dave Myszka is a Professor of Engineering Technology at the University of Dayton. Dave is a Registered Professional Mechanical Engineer in Ohio and is actively involved in applied research with industry. Dave received a B.S. and M.S. degrees in Mechanical Engineering from the State University of New York at Buffalo in 1985 and 1989, respectively. He also received an M.B.A. degree from the University of Dayton in 1996.

# **Motivating Students in an Introduction to Computing Course by Requiring Animated Solutions.**

**David Myszka  
Engineering Technology  
University of Dayton**

## **Abstract**

Technical computations is a vital skill for most engineering positions. Nearly every technical program requires some form of a computing course. Yet, students typically struggle with motivation and enthusiasm for these courses. In an attempt to provide extra incentive, students in an first-year technical computing course at the University of Dayton were required to animate their solution. It was observed that the students found great deal of satisfaction with achieving visual output. Students still performed classic introductory engineering calculations, but they were giving significant extra effort so they can watch their solution. This paper documents our strategy, gives examples of the computer programs, and reviews the encouraging results of student satisfaction surveys.

## **Introduction**

Numerical and computer solutions of technical problems have become a principle part of an engineering position. For decades, computer programming has been a fundamental part of the first year in both engineering and engineering technology curriculum. In the past, most institutions outsourced these courses to a computer science department. More recently, and in an effort to demonstrate applications to different technical disciplines, most institutions evolved to having these courses taught by engineering faculty (Gottfried, 1996).

Traditionally, high-level programming languages were used such as Fortran or C. With the development of computational tools, many institutions are moving towards using spreadsheets, VisualBasic for Applications or mathematical computation tools such as MATLAB, MATHCAD, and Maple (Howard, 2005).

In addition to learning computer programming as an essential skill, students can improve their logical reasoning and quantitative problem solving abilities. Additionally, this introductory class has the capability to introduce students to different engineering disciplines. Yet, several studies show that these courses often fall short of expectations (Gottfried, 1996, ). Today's students assert modest effort in these computing courses, and are not motivated to complete and debug programs (Jermann, 1996). Quite frankly, they find traditional programming to be dry, boring and irrelevant. Many techniques have been attempted to stimulate interest (Gottfried, 1997, Bowen, 2004, Said, 2004, Naraghi, 2001).

All engineering technology programs at the University of Dayton require an introductory technical computations course in the first year. Currently, VisualBasic for Applications (VBA), with Excel as a user interface is used. In an effort to address the concerns cited, a few exercises were assigned that required graphical, and animated outputs of the technical calculations. Using Excel, animations can be accomplished with little extra effort. It was observed that the students found great deal of satisfaction with achieving visual output. Students still can perform classic introductory calculations such as projectile motion, beam deflections, motion of a crane, etc. With this additional requirement, we witnessed students giving significant extra effort so they can watch a projectile move along its path, see a beam vibrating, and watch a “cherry picker” raising and lowering.

### **Motivation**

Some students seem naturally enthusiastic about learning, but most expect their instructors to inspire and stimulate them. Unfortunately, there is no single technique for motivating students. Many factors affect a given student's motivation to work and to learn (Sass, 1989) including interest in the subject matter, perception of its usefulness, general desire to achieve, self-confidence and self-esteem, as well as patience and persistence. And, of course, not all students are motivated by the same values.

In general, research has also shown that incentives, including grades, praise and privilege motivate students to learn (Tiberius, 1990) . However, studies also conclude that internal motivation is longer lasting and more self-directive than is external motivation (Lowman, 1990). This is the driving factor behind implementing the computational assignments that require animated output. The animations seemed impress the students by their own abilities. In most cases, success appeared to please the students and provide the internal motivation to achieve a correct solution.

### **Example Projects**

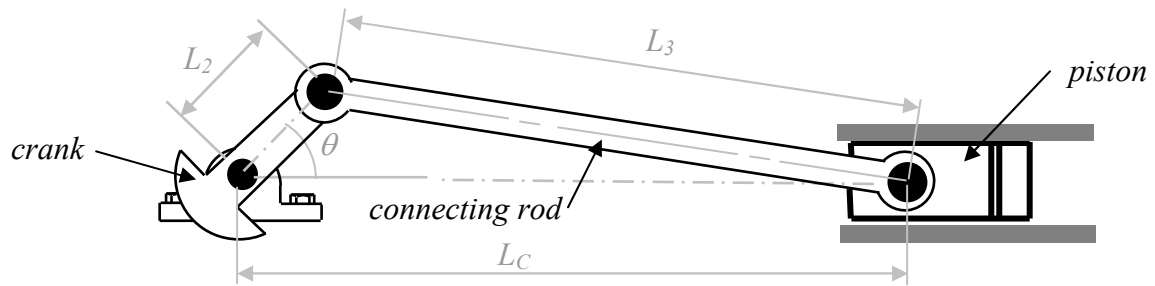
A few example projects are given below. The description given to the students always includes the same format, which includes background, underlying theory and the actual assignment. Also given below is a sample submission for each assignment.

#### **I. Slider-Crank Mechanism**

##### **A. Description Given to the Students**

###### ***Background:***

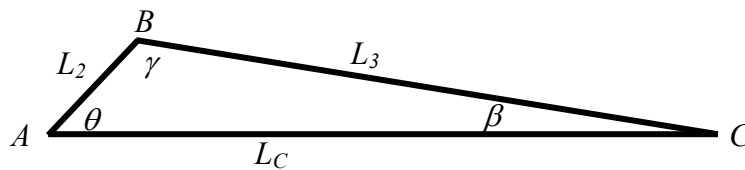
Many machines, most notably compressors, operate by imposing movement on a piston, in a linear manner, by rotating a crank. This common machine element is called a slider-crank mechanism and shown in figure 1. Mechanical design engineers often need to analyze a slider-crank, by determining the relationship between the angle of the crank and the corresponding position of the piston.



**Figure 1** Slider-crank mechanism

**Theory:**

Mathematically, this slider-crank mechanism forms a general triangle, as shown in figure 2.



**Figure 2** General triangle representing a slider-crank

This triangle can be analyzed through the fundamentals of trigonometry. The position coordinates of points *A*, *B* and *C* are:

$$\begin{array}{ll}
 A_x = 0 & A_y = 0 \\
 B_x = L_2 \cos \theta & B_y = L_2 \sin \theta \\
 C_x = L_2 \cos \theta + L_3 \cos \beta & C_y = 0
 \end{array}$$

Where:  $L_2$  = length of crank  
 $\theta$  = crank angle  
 $L_3$  = length of connecting rod

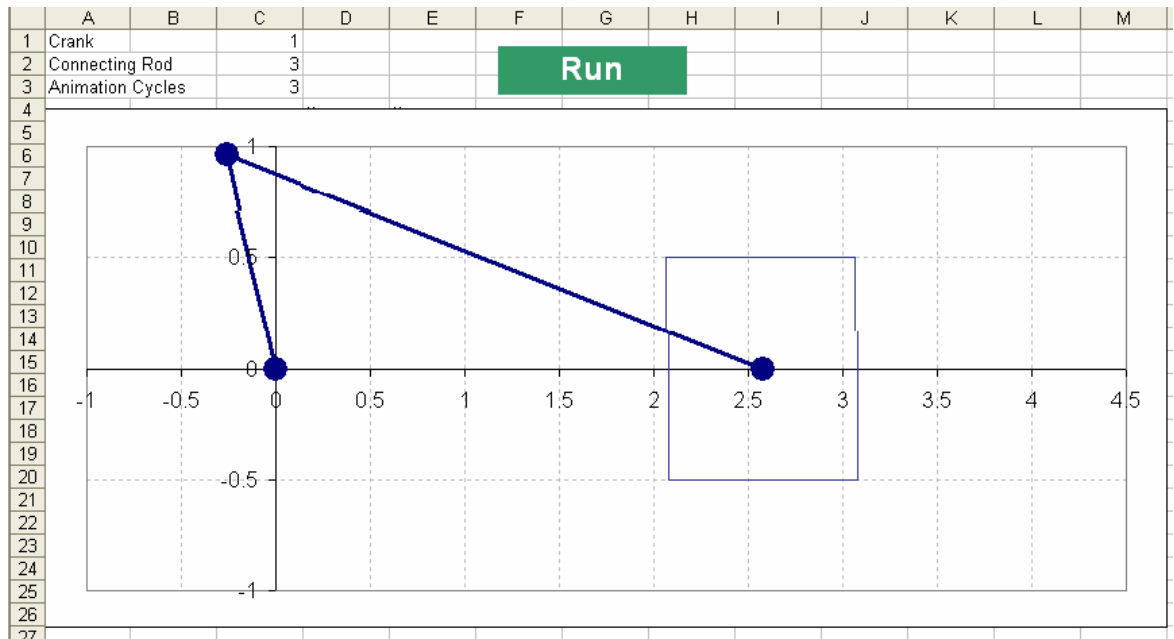
$$\beta = \sin^{-1} \left( \frac{L_2}{L_3} \sin \theta \right)$$

**Assignment:**

Create a VBA program will contains a loop that varies the crank angle,  $\theta$ , from  $0^0$  to  $360^0$  (or higher is you wish). Calculate and plot the position of points *A*, *B* and *C* at each loop step. If the update function is also used in the loop, this will have the effect of animating a slider-crank mechanism.

## B. Sample Submission

The submission from one student is shown in figure 3.



**Figure 3** Sample slider-crank submission

The VBA code to calculate and animate the sample shown in figure 3 is as follows.

```

Sub RunCrank()
    Pi = 3.14159
    crank = Sheet1.Cells(1, 3).Value
    Connect = Sheet1.Cells(2, 3).Value
    Ncycle = Sheet1.Cells(3, 3).Value
    For i = 1 To Ncycle * 360
        Angle = i * Pi / 180
        Sheet1.Cells(6, 4).Value = crank * Cos(Angle)
        Sheet1.Cells(6, 5).Value = crank * Sin(Angle)
        X = (crank * Sin(Angle) / Connect)
        beta = Atn(X / Sqr(-X * X + 1))
        Sheet1.Cells(7, 4).Value = crank * Cos(Angle) + Connect * Cos(beta)
        Sheet1.Calculate
    Next i
End Sub

```

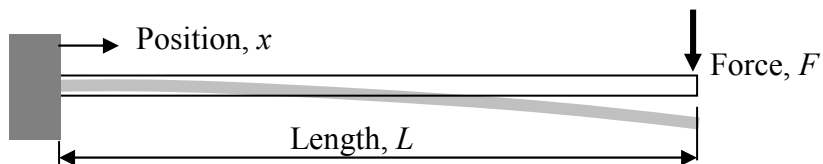
While the animation is appealing to the students and seems complex, the code is actually rather simple. It is seen that this solution involved formulating the technical calculations, and understanding a loop function. This assignment introduces first year students into a common analysis that many will encounter in a future course.

## II. Vibrating Beam

### A. Description Given to the Students

#### **Background:**

Mechanical design engineers often need to calculate the amount that different components of a machine will flex and deform under load, as the machine operates. Further, they must determine possible vibration effects. One common structural component is a bar, which is firmly held at one end, and loaded at the other as shown in figure 4. This structure is termed a “cantilever beam”.



**Figure 4** Cantilever beam

#### **Theory:**

The deflection,  $y$ , at any position,  $x$ , along the beam is predicted by the following equation:

$$y = \frac{-Fx^2}{6EI}(3L - x)$$

where:

$F$  = force applied to the end of the beam

$L$  = length of the beam

$E$  = elastic modulus, a material property:

=  $30 \times 10^6$  psi, for steel

=  $10.6 \times 10^6$  psi, for aluminum

$I$  = section inertia, a property of the cross-sectional shape of the beam:

=  $(\text{base})(\text{height})^3/12$ , for rectangular x-section

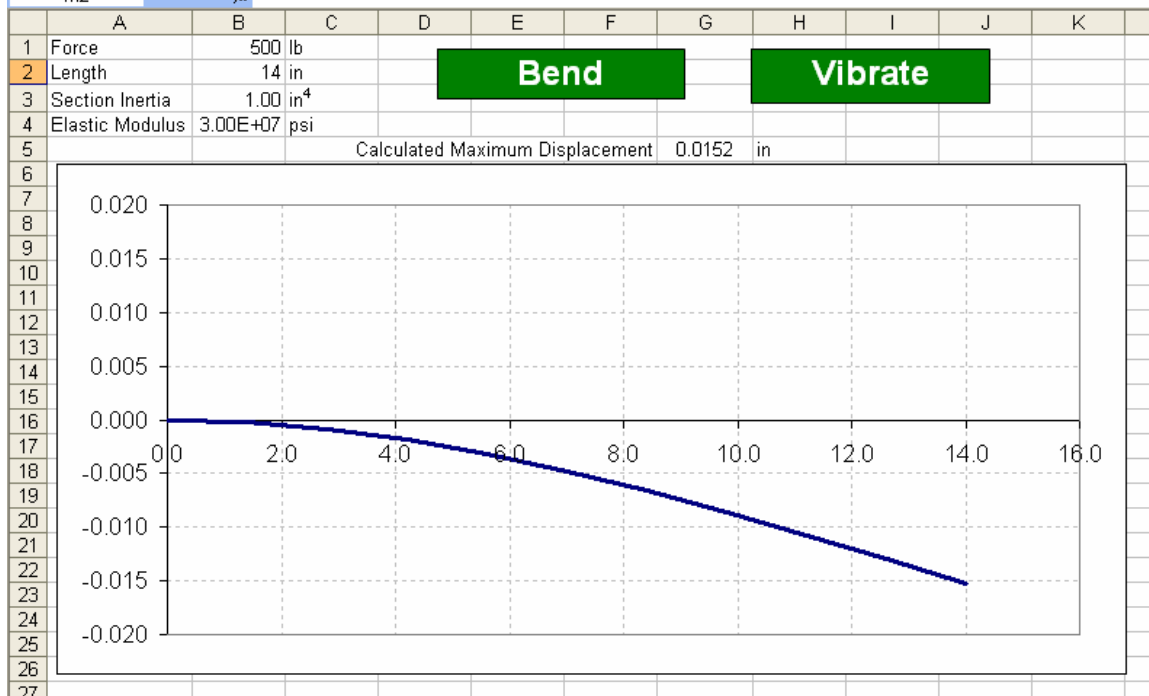
=  $(\text{diameter})^4/64$ , for circular x-section

#### **Assignment:**

Create a VBA program that will calculate the deflection at several positions along a beam, and place the results in two columns ( $x$ ,  $y$ ). Read the values for the force, length, section inertia and elastic modulus from cells. After calculation, plot the deflected shape. To animate the vibration, place the calculation in a loop, and multiply the deflection by the sine of the loop index.

### B. Sample Submission

The submission from one student is shown in figure 5.



**Figure 5** Sample beam submission

The VBA code to calculate and animate the sample shown in figure 5 is as follows.

```

Sub VibrateBeam()
Dim y(11) As Single
Pi = 3.14159265359

Force = Sheet1.Cells(1, 2)
Length = Sheet1.Cells(2, 2)
inertia = Sheet1.Cells(3, 2)
elasticity = Sheet1.Cells(4, 2)

For i = 0 To 10
    x = i * Length / 10
    Sheet1.Cells(7 + i, 1).Value = x
    y(i) = Force * x ^ 2 * (x - 3 * Length) / (6 * elasticity * inertia)
Next i

For i = 0 To 1500
    ScaleFactor = Cos(Pi * i / 150)
    For j = 0 To 10
        Sheet1.Cells(7 + j, 2).Value = ScaleFactor * y(j)
    Next j
    Sheet1.Calculate
Next i

End Sub

```

Again, the animation of a vibrating bar is appealing to the students and seems complex, the code is actually rather simple. The solution involved formulating the technical calculations, and understanding an array and loop function. Also, this assignment introduces first year students into a common analysis that many will encounter in a future course.

### III. Digital Circuits

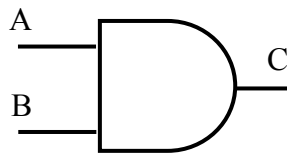
#### A. Description Given to the Students

##### **Background:**

Electronic design engineers often need to use digital logic to create circuits for different applications. One such application is a home security system. In such a system, sensors are placed at windows and doors to detect open or closed configuration. The system could also include a motion detector. Of course, the owner would want a master on/off switch. The circuit design engineer must design a circuit that would decide the circumstances that would sound the alarm.

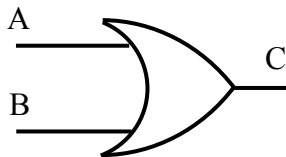
##### **Theory:**

A digital circuit is comprised of a series of combinational logic devices, or gates. These digital devices convert binary input into binary outputs based on the rules of logic. One of the most common is the AND gate, which is skematically shown in figure 6. The AND gate will be true (1) if and only if all input signals are true (1). Otherwise, the output is false (0).



**Figure 6** And gate

Another of the most common is the OR gate, which is skematically shown in figure 7. The OR gate will output true (1) if any of the input signals are true (1). Otherwise, the output is false (0).



**Figure 7** And gate

Many circuits can be constructed from these two elementary gates.

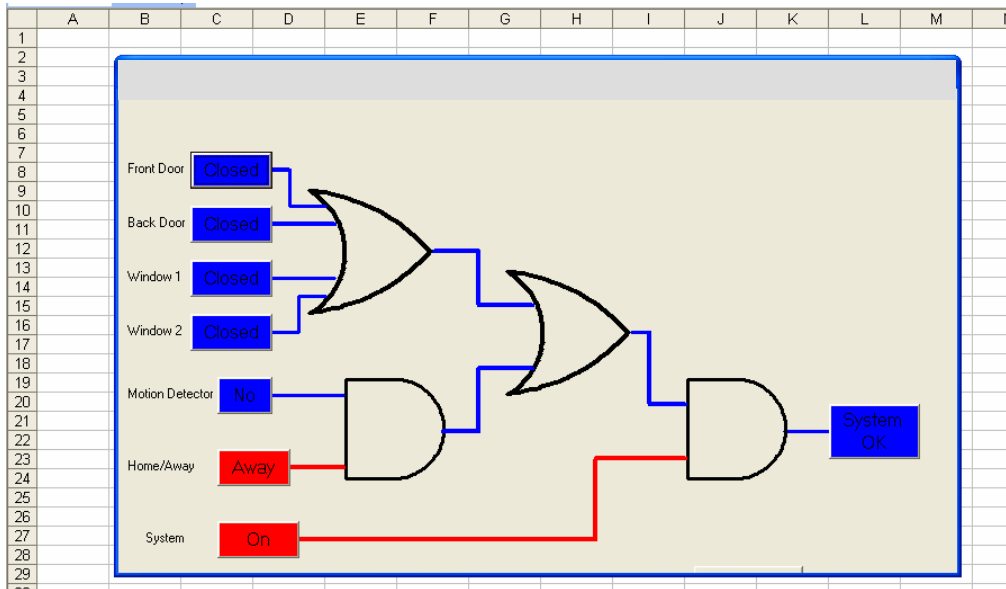


**Assignment:**

Create a VBA program that will display the logic involved in a home security system. The system should include sensors at doors and windows (open = true, closed = false). It should include a motion detector. And a master on/off switch.

**B. Sample Submission**

The submission from one student is shown in figure 8.



**Figure 8** Sample home security submission

A portion of the VBA code to calculate and animate the sample shown in figure 6 is as follows.

```
Sub calculate()  
  If In1 = True Or In2 = True Or In3 = True Or In4 = True Then  
    LineOut1a.BorderColor = &HFF& 'red  
    LineOut1c.BorderColor = &HFF& 'red  
  Else  
    LineOut1a.BorderColor = &HFF0000 'blue  
    LineOut1b.BorderColor = &HFF0000 'blue  
  End If  
  
  If In1 = True Or In2 = True Or In3 = True Or In4 = True Then  
    LineOut1a.BorderColor = &HFF& 'red  
    LineOut1c.BorderColor = &HFF& 'red  
  Else  
    LineOut1a.BorderColor = &HFF0000 'blue  
    LineOut1c.BorderColor = &HFF0000 'blue  
  End If  
End Sub
```

Etc.

While the code was straightforward, it involved significant usage of conditional statements. Additionally, this assignment introduces first year students into a digital circuit logic, that many will encounter in a future course.

## Results

Instructor observations that students did place extra effort into the assignments that required an animated solution. In an attempt to quantify that observation, the students completed a survey at the conclusion of the semester. The students ( $N = 32$ ) were asked to indicate the factors that motivated them in diligently completing an assignment. A standard 5-point Likert scale was used (5-strongly agree, 1-strongly disagree). The results are shown in table 1.

**Table 1:** Results of student survey

<i>Inquiry</i>	<i>Avg.</i>	<i>Std. Dev.</i>
1. I am more likely to place extra effort into my computer assignment when I know it will be thoroughly graded.	4.06	0.95
2. I am more likely to place extra effort into my computer assignment when I know a similar problem will appear on a test.	4.34	0.60
3. I am more likely to place extra effort into my computer assignment when the correct answer is given.	3.84	0.95
4. I am more likely to place extra effort into my computer assignment when the results will be displayed in a graph.	3.25	0.76
5. I am more likely to place extra effort into my computer assignment when the results will be animated.	4.16	0.88
6. I am more likely to place extra effort into my computer assignment when I understand the significance of such calculations.	4.09	0.73
7. I am more likely to place extra effort into my computer assignment when the calculations are directly related to my declared major.	4.06	0.80
8. I am more likely to place extra effort into my computer assignment when the calculations I am working with a group of students.	3.94	1.06
9. I am more likely to place extra effort into my computer assignment when the results will be used in a subsequent assignments.	4.09	0.73
10. I am more likely to place extra effort into my computer assignment if I am asked to present it to the rest of the class.	4.03	0.93

Clearly, the greatest motivation factor for attentively completing an assignment was the possibility of a similar problem being on a test. The second most important factor was creating an animated solution. This substantiated our observations. However, it was found that understanding the significance of the calculations, and completing computations related to their major were nearly as important. In response to this finding,

all assignments were formatted as shown in the examples above. Each project provides background information that attempts to discuss the relevance of the required computations. Additionally, the background specifies the technical discipline that is mostly responsible for that type of analysis.

Surprisingly, merely plotting a solution on a graph, which provides a visual representation of the solution, proved to be the least motivating factor. Prior to the survey, the instructors felt that graphing a solution provided significant motivation.

## **Conclusions**

Today's college students have been raised in a animated world, filled with video games, web sites, computer generated movies, and amazing special effects. This generation of students seem to struggle with perseverance when solving numerical problems. For most, little motivation is gained by achieving a numerical solution.

The few projects documented in this paper achieve the traditional learning outcomes of a technical computations course. They lead the students through important engineering calculations. Additionally, they provided some internal incentive, as the students find satisfaction in "watching" their solution. These projects give them a taste of creating the animations they regularly experience in the movies, web sites and their video games. Since the survey was administered, several more projects have been written that require animated solutions.

Also, in addressing the survey results, more emphasis has been placed on the relevance of the calculations. Each project statement attempts to thoroughly describe the need for the calculations, and the technical professional who would perform the analysis. The instructor also addresses the relevance issue in much greater detail.

Further, the survey reveals that students are motivated if the calculations involve their chosen degree major. Since this is an interdisciplinary course, the projects could not be tailored to one focus. Yet, in generating the projects, they are equally distributed across mechanical, manufacturing, electronic, computer and industrial engineering technology applications.

This survey will be conducted in future course offerings to verify the motivation factors. In addition, questions will be added that relate to the achievement of those motivation factors with the newly created projects.

## **References**

Bowen, J. "Motivating Civil Engineering Students to Learn Computer Programming with a Structural Design Project". *Proceedings of the ASEE Annual Conference and Exposition*, 2004, Session 1793.

- Gottfried, B. "Should Computer Programming Be Taught to all First-year Engineering Students". *Proceedings of the ASEE Annual Conference and Exposition*, 1996, Session 1253.
- Gottfried, B. "Teaching Computer Programming Effectively Using Active Learning". *Proceedings of the ASEE Annual Conference and Exposition*, 1997, Session 1045.
- Howard, W., Musto, J. "Computer Skills in an MET Curriculum". *Proceedings of the ASEE Annual Conference and Exposition*, 2005, Session 1348.
- Jermann, W. "The Freshman Programming Course: A New Direction". *Proceedings of the ASEE Annual Conference and Exposition*, 1996, Session 1532.
- Lowman, J. "Promoting Motivation and Learning." *College Teaching*, 1990, 38(4), 136-39
- McKeachie, W. J. *Teaching Tips*. 11/E. Heath, 2002.
- Naraghi, M., Litkouhi, B. "An Effective Approach for Teaching Computer Programming to Freshman Engineering Students". *Proceedings of the ASEE Annual Conference and Exposition*, 2001, Session 2520.
- Said, H., Khna, F. "Towards Using Problem-Based Learning in Teaching Computer Programming". *Proceedings of the ASEE Annual Conference and Exposition*, 2004, Session 3260.
- Sass, E. J. "Motivation in the College Classroom: What Students Tell Us." *Teaching of Psychology*, 1989, 16(2), 86-88.
- Tiberius, R. G. *Small Group Teaching: A Trouble-Shooting Guide*. Toronto: Ontario Institute for Studies in Education Press, 1990

## **Biographical Information**

DAVID MYSZKA

Dave Myszka is a Professor of Engineering Technology at the University of Dayton. Dave is a Registered Professional Mechanical Engineer in Ohio and is actively involved in applied research with industry. Dave received a B.S. and M.S. degrees in Mechanical Engineering from the State University of New York at Buffalo in 1985 and 1989, respectively. He also received an M.B.A. degree from the University of Dayton in 1996.