AC 2010-749: M-MODEL: AN ONLINE TOOL FOR PROMOTING STUDENT PROBLEM SOLVING UTILIZING MENTAL MODELS

Edward Anderson, Texas Tech University

Edward Anderson is a Texas Tech University, Department of Mechanical Engineering Professor. He has also served as the Ray Butler Distinguished Educator and received the Chancellor's Council Distinguished Teacher and the Texas Piper Professor awards. He has written two engineering textbooks and numerous textbook websites. His research focuses on application of technology to learning and how to measure and promote higher-order cognitions using technology. During 2009/10, he served as a Distinguished Visiting Professor at the United States Air force Academy.

Roman Taraban, Texas Tech University

M-MODEL: An Online Tool for Promoting Student Problem Solving Utilizing Mental Models

Abstract:

Students learn to solve problems by developing mental models of the problem. Although these models are many and diverse, a common one used in engineering education consists of identifying the known and unknown variables, construction of a graphical problem representation, and developing a mathematical model representing the two preceding steps. This is particularly the case for courses in physics, mechanics, and electrical circuits. M-MODEL is a computer-based implementation of this approach to problem-solving. It requires users to build the known/unknown, graphical (free-body diagram in this paper), and mathematical models of a problem. Once the student creates a complete model, M-MODEL checks it for errors such as proper number of graphical elements, naming of variables, and equation formatting. These checks also provide users with feedback that can be used to correct or improve their models. Once users are satisfied with their models, M-MODEL proceeds to solve their equations as well as display the correct solution for users to compare to their models. M-MODEL also provides a tool that individual authors can use to prepare problem models.

This paper discusses the features of M-MODEL as applied to an Engineering Statics course. It also discusses how it may be used to encourage students to develop mental model approaches to problem solving.

Introduction:

Students solve problems by constructing mental representations or mental models of the problem. These models take many forms such as graphical, mathematical, flow charts, process steps, and schematics to mention a few. As pointed out by Norman[10], these models can be contradictory, incomplete, superstitious, erroneous, and unstable, while varying in time. It is the task of the educator to help students learn how to form accurate and useful mental models and apply them to knowledge domain problems. M-MODEL is a computer-based tool that permits mechanics and engineering educators to develop problems using the principle mental models of the discipline in a consistent and flexible manner. This paper describes the user and problem author environments, the philosophy behind M-MODEL, and some of the pedagogues embedded in it.

Several engineering problem solving models or schema have been reported recently. These include the Wankat and Oreovicz[19] problem solving strategy, McMaster problem solving program of Woods[20] and Woods, et al.[21], Gray and colleague's[6] structured approach to problem solving, Mettes and associates[9] Systematic Approach to Solving Problems, and Litzinger, et al.'s[8] Integrated Problem Solving Model. The Wankat and Oreovicz strategy divides problem solving into definite steps including motivation, exploration, and reflection as well as the more common define, plan, execute and check steps. The McMaster problem solving program uses a structure similar to that of Wankat and Oreovicz and implements it across entire curricula. Gray's structured approach emphasizes pattern-matching that starts with a small number of general equations that students reduce to fit a given situation. The Mettes problem solving schema is based upon a flow chart of problem solving steps and a constructionist approach to learning. Litzinger's integrated model emphasizes problem representation and the conversion from one representation (say problem statement) to another (say graphical).

The problem solving define, plan, and execute steps are the common thread among these various models. In mechanics disciplines, these steps take the form of free-body diagram (FBD) development, listing of the given (known) and find (unknown) variables, creating a mathematical model consistent with the FBD, and final answer production. As pointed out by Gray and Costanzo[7], the current trend in mechanics is to deemphasize the final answer production step and leave this to computational software. Mechanics educators are now emphasizing model building rather than computational procedures. Four mental models are commonly used to build problem solutions. These are: problem statement, graphical representation such as a FBD, given/find representation typically in the form of lists, and a system of equations that will produce the final answers. These four mental models represent the core of mechanics problem solving and have a long standing tradition in mechanics education.

The newest trend in mechanics education is the application of computer technology to teach students, engage them in the learning process, and to help them understand mechanics concepts and principles. These are many and varied. They are perhaps best illustrated by the works of Gramoll[3,4], Dollar and Stief[2], Stief and Dollar[15,16], Philpot[11], Philpot and Hall[12], Stanley[14], and Gray and Costanzo[5] to list a few. Many of these are similar to traditional textbook presentations with exceptions such as interactive examples, audio/video lectures, homework sets with immediate feedback, virtual experiments, and interactive animations to develop conceptual understanding. This approach to learning problem solving is based upon examples and homework problem sets which is fairly traditional. Problem interactivity has been added to keep the student engaged with the problem. Hints, intelligent coaching, instantaneous feedback, and intelligent correcting have been incorporated by many of these authors. But, they rely upon click-on-object, drop and drag, pair matching, multi-choice answers, and short answers (usually numerical) for user inputs and traversing the basic problem structure. They tend to be somewhat inflexible in that users must use notations, axis systems, vector directions, equation ordering and etc. as prescribed by the problem designer rather than allowing students to make choices and decisions on their own.

The Andes problem solving system for classical physics developed by VanLehn, et al.[18] and implemented at the United States Naval Academy by Schultz, et al.[13] is based upon a Baysian network representation of a problem. This system allows considerable flexibility and generates solutions, immediate feedback, and help comments based upon the path traversed by the user through this network. Hence, the user who elects to use one set of notations will be coached through the problem and produce a correct answer for that notation set just as the user who chooses to use some other notation system. This approach encourages students to think through the solution, plan their approach, and develop in-depth problem solving skills rather than charging directly and often blindly into and through the problem solution. Andes utilizes four mental models, problem statement, graphic representation, variables lists, and mathematical model, and requires users to develop each of these mental models (graphical representation is optional). Andes includes an equation solving tool although users can also solve the equations off-line. A research project conducted on some 330 students approximately one-half of whom were in a control group, resulted in a 3% (1/3 letter grade) student performance improvement on departmental pencil and paper examinations by students who did Andes homework rather than traditional homework. Anecdotal results from Andes users (Schulze, et al.[13]) indicate that students are initially reluctant to carefully define their variables, some students ask for help on almost every step of a problem solution, giving effective hints and help is very difficult, and analysis of actions logs reveal that students do not understand physics as well as might be thought.

M-MODEL utilizes the same four mental models as Andes and requires users to fully develop their graphical, variables, and mathematical representations from the problem statement. Although most users will develop their representations in this order, it is not required and students can proceed however they deem appropriate. But, all user representations must be completed before a correct solution is possible. Users have complete freedom in naming their variables, orienting their FBD vectors and coordinate systems, selecting their units, and etc. as they set up their solution. These choices are graded against the problem designer's expectations and final answers. M-MODEL is therefore an extremely versatile system that gives students considerable freedom in developing their problem solution and encourages them to utilize in-depth problem solving skills and high-order cognitions.

M-MODEL Philosophy:

M-MODEL was conceived as a tool students can use to practice and develop their problem solving skills as well as to allow sufficient flexibility that varied, but correct, problem solution paths are possible. This latter objective is important in that mechanics courses are typically those courses that begin the transition from well-framed problems to the more ill-defined engineering and design problems. It is also important that students learn how to formulate problems and that correct, but different, answers depend upon that formulation.

M-MODEL was also designed to require users to use all four mental models common to the current problem solving models in the engineering educational literature. These mental models have a long standing tradition in mechanics education and are familiar to engineering educators. M-MODEL also focuses on the model building process and leaves the computational details to an embedded equation solver. This feature is intended to channel students away from the rush to simply write equations, substitute values, and produce answers. Rather, students must carefully build complete models before a final solution is possible. Focusing on model building is also the current trend in engineering and mechanics education.

M-MODEL is also intended to give problem developers a tool that they can use to develop their own problems and homework sets. It is also designed to reduce the task of grading student solutions. It grades many student mistakes and misconceptions and records these grades when implemented on a database and active page server. The software also records user activities, sequences, and time-on-task for research and verification purposes.

M-MODEL User Interface:

The initial user screen is shown in Figure 1. This screen is divided into 5 different areas: problem statement, graphical representation, variables listing, pre- and post-calculations, and equation system. The problem statement is presented in the upper, left portion of the screen. Problems can contain up to two random parameters. This example contains one, the gymnast's weight. Users are encouraged to select a free-body from the problem statement diagram by clicking on the appropriate object. Users are not required to do this at the beginning, but ultimately they must select a free-body and develop a FBD. Points are deducted from a student's score and an error message appears if an incorrect free-body is selected. Students can complete with their solution once the correct free-body has been selected.

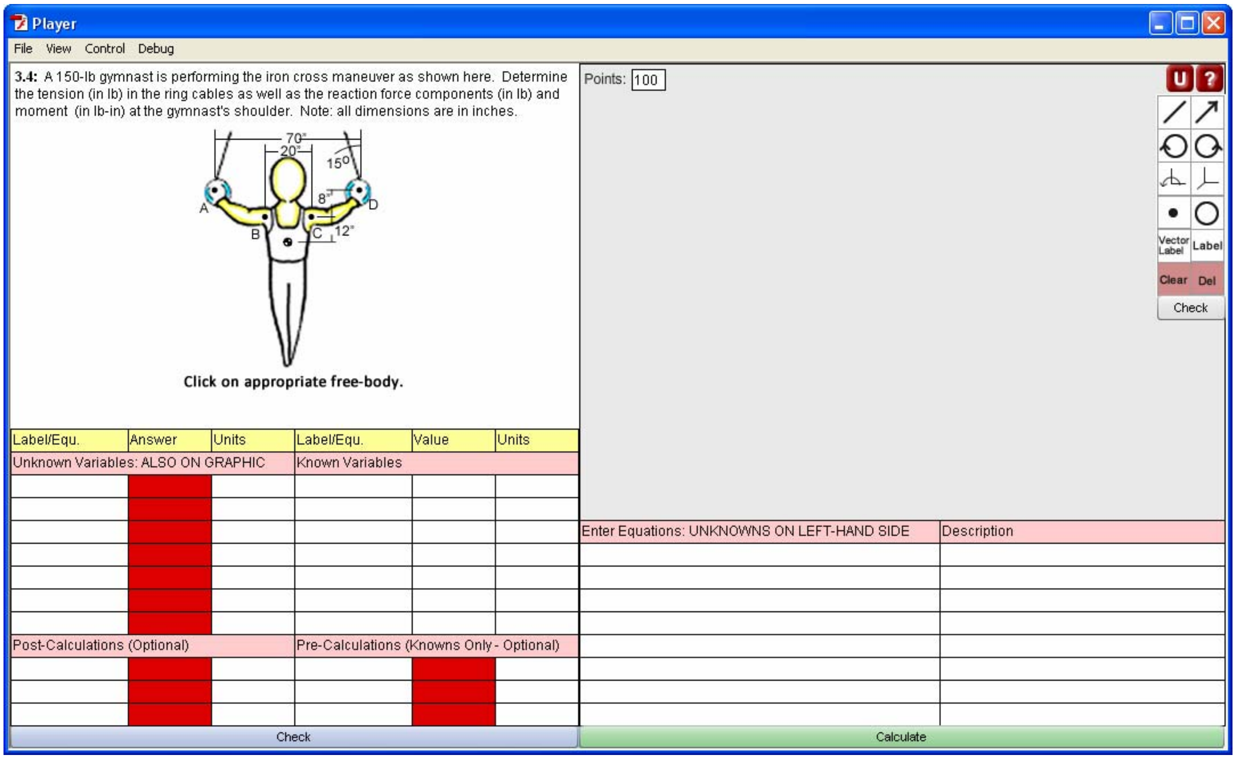

Figure 1: Initial User Interface

Once a user selects the correct free-body, it appears in the graphics representation window of the upper, right-hand screen where the user can use the tools in the toolbox to complete the FBD. These tools include: lines, arrows, clockwise and counter-clockwise moments, two- and three-dimension axis, points, circles, vector (bold) labels, plain labels, screen clearing, and object deleting. Any time a tool is selected by the user, a pop-up screen with instructions on the tool's usage appears. Some of the objects produced by the tools (e.g., axis and moments) can also be rotated as appropriate for the problem. Users click the "Check" button below the tool box to determine if their FBD is complete. An example of a completed FBD is shown in Figure 2.

Problem designers set the minimum number of graphical objects users must add to the FBD. This determines the detail that is expected for correctly completing a FBD. This may be as simple as 3 arrows, 1 moment symbol, and 1 two-dimensional axis for the example of Figure 2 or as complicated as shown in Figure 2 with additional lines and labels.

The graphical mental model window includes a grade display and two help buttons. Users begin with 100 points. This number is reduced whenever the user makes a mistake. This reduction depends upon the significance of the mistake. The deduction is set to be higher for a major error like selecting an incorrect free-body and less for a minor mistake like indistinct labels. The magnitude of the deducted points is at the discretion of the problem designer. The "U" button in the upper right-hand of this window activates a pop-up list of the unit symbols in M-MODEL.

The “?” button activates another pop-up with an abbreviated set of instructions as a quick reference for users. Full user instructions are available in a separate file.

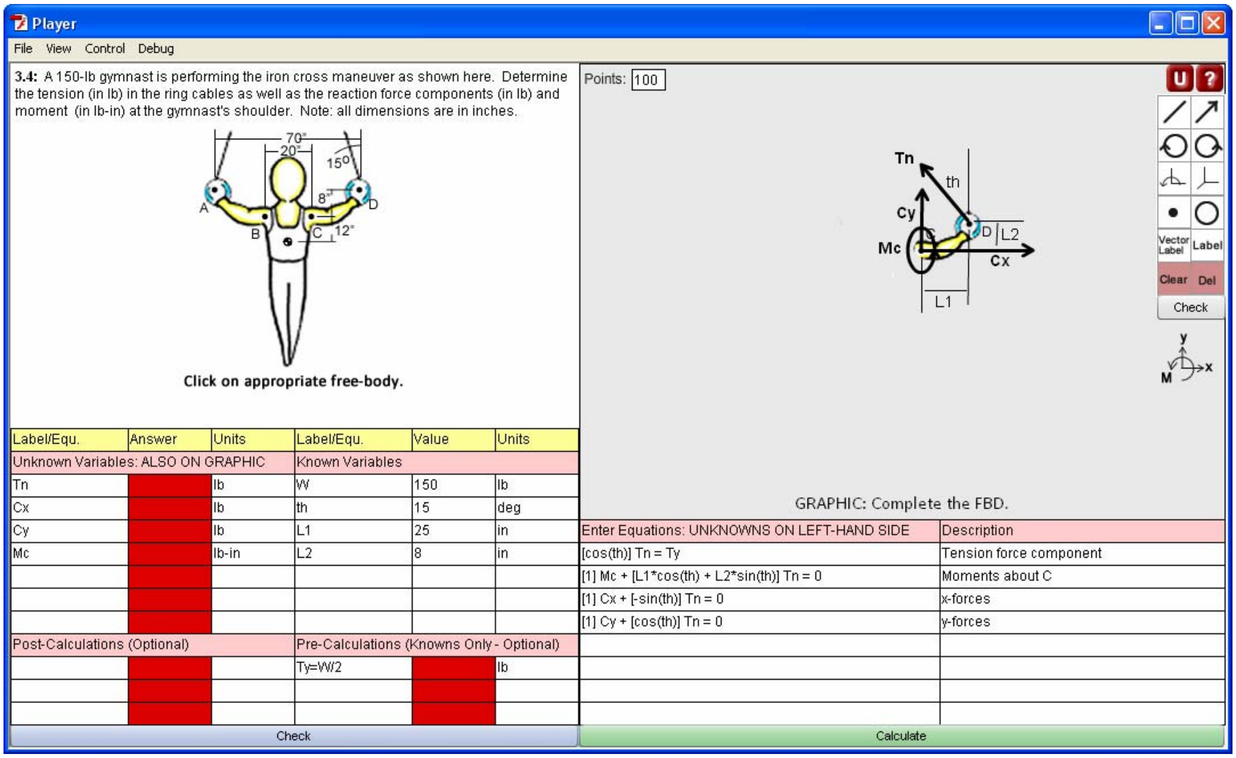

Figure 2: User screen prior to completing calculation

The left-hand side of the user interface also contains a variables list and calculations section in the lower quadrant as shown in Figure 2. The variables list is divided into lists of the unknown variables and known variables. These are completed by entering the labels and units for the known and unknown variables. The value of the known variables must also be entered by the user. The calculated value of the unknown variables is displayed once the user completes her model and clicks the “Calculate” button to the right. Once the “Check” button is clicked, the unknown variable labels are checked against those on the FBD (the FBD can contain more labels then those on the unknown variable list), checks all labels to insure they are distinctly named, and checks all units against a problem designer approved list. Error messages are displayed and points deducted if any of these errors occur. These error messages provide immediate feedback and give the user the opportunity to make corrections prior to completing the full solution.

Pre-calculation equations and units of the resulting pre-calculation variables are entered in the “Pre-Calculations” list. Only constants and known variables can be used to build the pre-calculation equations. These are entered in equation form with the pre-calculation variable label on the left-hand side of the equals sign. Typical examples of pre-calculations are items like moments of inertia and areas. The pre-calculation equations are the first to be evaluated when the “Calculate” or “Check” button is clicked. The results of these equations are displayed at this time and the pre-calculation variables become available for the subsequent evaluation of the system of equations in the lower, right-hand section of the user interface and the post-calculation equations. The final user screen illustrating the result of these calculations with a single pre-calculation is shown in Figure 3.

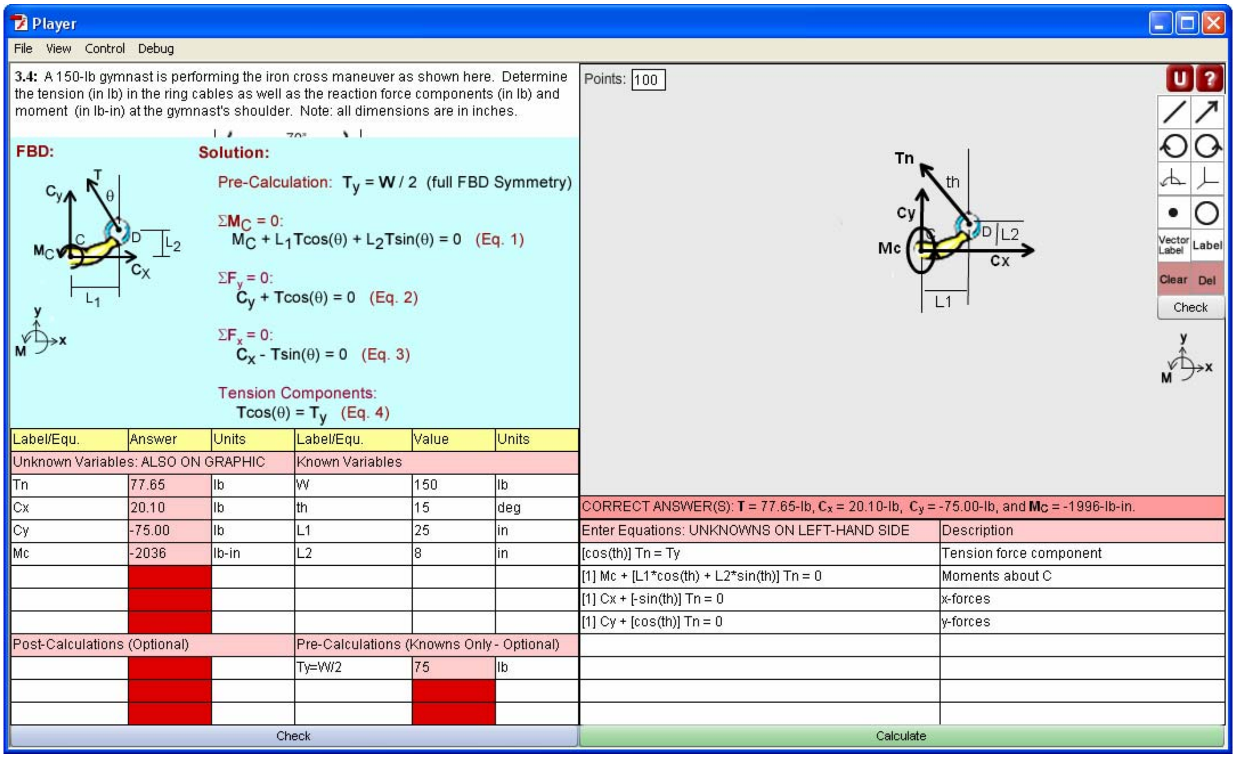

Figure 3: Final User Screen

The "Post-Calculations" section is organized in the same manner as the "Pre-Calculations" section. The only difference is that post-calculations are done once the unknown variables have been determined by solving the pre-calculation and equation system equations. At this point all variables are available for the post-calculation equations and they may be evaluated. A typical example of a post-calculation is an axial stress calculated from an unknown internal force and section area.

The last section to be completed by the user prior to calculating final answers is the "Equations" section in the lower, right-hand quadrant of Figure 2. This system of equations is commonly, but not necessarily, the equations of force and moment equilibrium. This is divided into two lists: equations and comments. Comments are optional. Equations are entered using the following simple rules:

- Unknown variables are always on the left-hand side of the equation and each unknown variable can only appear once per equation.
- Unknown variable coefficients and expressions precede the variable name and are delimited by braces (see Figure 3).
- Unknown variable terms are always added, never subtracted (subtraction is accomplished by negating the variable's coefficient as shown in Figure 3).

Terms involving only constants, known variables, and pre-calculation variables are entered into these equations as appropriate. The equation solver includes the *, /, +, and – operators. The equation solver also includes the following functions: sin, cos, tan, asin, acos, atan, pow (x^y), and ln (natural logarithm). These are used to build the terms and coefficients of the pre-calculation, post-calculation, and equation system equations.

Equations entered in this section must be independent, linear equations with the number of equations matching the number of unknown variables. Any non-linear calculations (e.g., diameter of a circle given the area) can be done in the "Pre- and Post-Calculation" sections. When the "Calculate" button is clicked, several equation format checks are done, error messages displayed, and points deducted if appropriate. The system of equations is also checked at this time to insure that they are independent. The equations system and post-calculation equations are then solved. Next, the values of the unknown variables and post-calculation variables are compared against a list of correct answers provided by the problem designer. If the problem statement includes random parameters, the problem designer must provide correct answer equations for each of the unknown and post-calculation variables that only include constants and the random variables. User answers are considered incorrect if they are not within +/- 1% of the designer's answers. Incorrect user answers are highlighted and the user can proceed to edit any item on the screen and recalculate their answers. The user's opportunity to revise a solution based upon feedback is known to achieve deep, lasting learning (Suskie[17]).

M-MODEL Authoring Tool:

The first step in creating an M-MODEL problem is drawing the problem statement, free-body, and solution graphics shown in Figure 3. The free-body graphic is normally a copy of the problem statement graphic with all but the free-body object removed or erased. Any graphics editor that produces jpg, gif, or png graphic files can be used for this purpose. Authors save these graphic files in a folder or directory of their choice. The authoring tool shown in Fig. 4 can then be used to create a new M-MODEL problem or edit an existing problem.

The first items on this form are the names and locations of the three graphics files. The problem statement is then entered in the Question field using HTML markup tags for formatting. These tags are: <b> - bold font, <i> - italic font, <sub> - subscript, <sup> - superscript, <p> - paragraph break, and <sym> - symbol font. Up to two random variables named var1 and var2 may be inserted anywhere in the question statement. The random variable minimum value, maximum value, and step size dictate the range and division of the random variables and are entered in the appropriate fields of Figure 4. The axis system (2- or 3-dimensional) is determined by entering 2 or 3, respectively, in the Axis field. The minimum number of the various graphical objects is then entered into appropriate fields. Acceptable units, separated by #'s, are entered in the Units field. Point deductions for major and minor errors are entered into their respective fields. The title for the graphic mental model panel is entered into the Graphic Title field. Correct answer equations for all the unknown and post-calculation variables, separated by #'s, are entered in the Answer Equations field. These answers are given labels, such as ans1, using standard equation notation. These equations can only use expressions involving constants and any random variables in the question statement. The correct answers statement, including any correct answers from the preceding field, is entered in the last field. These correct answers are denoted by their label. This form is saved as an xml file in the same directory and with the same file name as the problem statement graphic file.

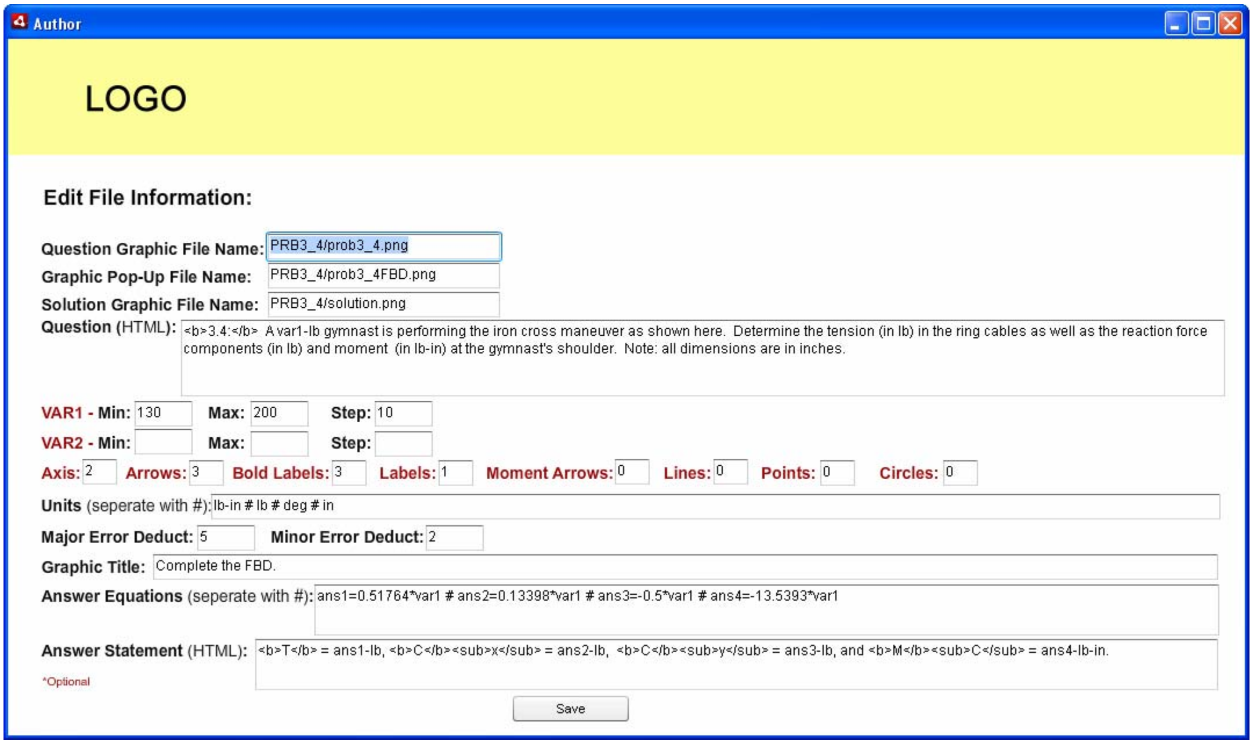

Figure 4: Problem Authors Form

M-MODEL Pedagogues:

M-MODEL is a non-sequential problem solving tool that encourages students to build their own problem mental models with as few restrictions as possible. Although users must complete four of the six models (pre- and post-calculations are optional), they can be completed in any order. They can also be altered before final solution as one model provides further insight into another model. For example, users often add or remove variables as they are writing their equations or refining their graphic model. These cognitions fall under the Analyze (breaking down material or tasks into constituent elements) and Evaluate (making judgments using standards and criteria) classifications of Bloom's taxonomy (Bloom[1]). Both are at the higher-order cognitions end of Bloom's taxonomy.

M-MODEL allows users to set up and solve a problem in their own terms using solution procedures of their own creation. For example, the problem of Figures 1-3 can be correctly solved by considering *Ty* a pre-calculation variable or an unknown variable. In the latter case, the user must include an additional equation in the equation section and the *Ty* label in the graphics panel. Users may also elect to not use *Ty* at all, but rather to replace it with W/2 where appropriate. Other users may elect to use some or all numerical values in lieu of variable labels and values. All of these choices will produce correct answers. User's must then Create (Producing alternatives or reorganizing materials in new ways) solutions which is the highest-order cognition in Bloom's taxonomy.

Although M-MODEL promotes procedural and higher-order cognitions, it is not without its penalties. First, students cannot produce correct answers without a through and detailed set of models, and often they need to refine or rebuild their models as their understanding of the problem deepens. This entails additional work on the part of the student which frequently meets with objection. Users need to solve 2-3 problems to become comfortable with the interface. This learning curve can interfere with their learning the content material and may frustrate them. Some of this extra effort is recovered by the equations solver which saves some time. The automatic solving of the equations and lack of computational practice can be problematic during examinations if numerical answers are heavily weighted.

Conclusions:

M-MODEL is a flexible, computer-based problem solving tool based on the problem statement, graphical, given/find, pre-calculations, post-calculations and equations mental representations of a problem. It is consistent with the model building pedagogue of current mechanics education. Individual problem creators can program their own problems with minimal effort using the M-MODEL authoring tool.

Its design encourages higher-order cognitions required to bridge from linear, simple problems to more ill-defined problems on the engineering intellectual development spectrum. The flexibility of its problem solving procedure challenges students to think more deeply about problems and helps them develop the confidence they need to apply their own approach to a problem. This tool also removes the burden of prerequisite computational procedures so that learners can focus upon model building which is so critical to solving mechanics problems.

A project to measure student perceptions and performance gains was initiated at the time this paper was being written. But, in view of its similarity to the Andes project of physics (VanLehn[18]), similar perceptions and performance gains might be expected. The authors intend to report on these results at the time of the paper's presentation, if they are available.

References:

1. Bloom, B. S., and D. R. Krathwohl. 1956. *Taxonomy of educational objectives: The classification of educational goals*. New York, NY: Longmans, Green, and Co.
2. Dollár, A. and Steif. P. S. 2003. "Learning modules for the statics classroom", *Proc*. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*. Nashville, TN.
3. Gramoll, K. 2007. A web-based electronic book (EBOOK) for solid mechanics. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*, Honolulu, HI.
4. Gramoll, K. 2009. eCourses. http://www.ecourses.ou.edu.
5. Gray, G. L. and Costanzo. F. 1999. Interactive dynamics: A collaborative approach to learning undergraduate dynamics . *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*. Charlotte, NC.
6. Gray, G. L., Costanzo, F., & Plesha, M. E. 2005. Problem solving in statics and dynamics: A proposal for a structured approach. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*, Portland, OR.
7. Gray, G.L. and Costanzo, F. 2007. Toward a new approach to teaching problem solving in dynamics. in *Learning to solve complex scientific problems*. Jonassen, D.H., Ed. New York, NY. Taylor and Francis group.
8. Litzinger, T., P. Van Meter, M. Wright, and J. Kulikowich. 2006. A cognitive study of modeling during problem solving. *Proceedings of the American Society for Engineering Education Annual Conference &*

Exposition. Chicago, IL.
9. Mettes, C. T. C. W., A. Pilot, H. J. Roossink, and H. Kramers-Pals. 1981. Teaching and learning problem solving in science. *Journal of Chemical Education 58* (1): 51-55.
10. Norman, D.A., 1983. Some observations on mental models. in *Mental Models*, Gentner, D. and Stevens, A.L., Eds. New York, NY. Taylor and Francis Group.
11. Philpot, T.A. 2008. *Mechanics of materials: an integrated learning system*. New York, NY. Wiley.
12. Philpot, T.A. and Hall, R.H. 2005. The role of MDSolids in international mechanics of materials education, *International Journal of Engineering Education, 21* (2):
13. Schultz, K.G., Shelby, R.N., Treacy, D.J., Wintersgill, M.C., VanLehn, K., and Gertner, A. 2000. Andes: an intelligent tutor for classical physics. *The Journal of Electronic Publishing, 6* (1).
14. Stanley, R. 2009. A way to increase the engineering student's qualitative understanding of particle kinematics and kinetics by utilizing interactive web-based animation software. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*. Austin, TX.
15. Stief, P. and Dollar, A. 2007. An interactive web-based statics course. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*, Honolulu, HI.
16. Stief, P.S. and A. Dollar, 2009. Study of usage patterns and learning gains in a web-based interactive static course. *Journal of Engineering Education*, 98(4): 321-334.
17. Suskie, L., 2009. *Assessment student learning: a common sense guide, 2nd ed.* San Francisco, CA: Jossey-Bass.
18. VanLehn, K., C. Lynch, K. Schulze, J.A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. 2005. The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education*, 15(3): 147-204.
19. Wankat, P. C. and F. S. Oreovicz. 1993. *Teaching engineering*. New York, NY: McGraw-Hill.
20. Woods, D. R. 2000. An evidence-based strategy for problem solving. *Journal of Engineering Education,* 89 (3): 443-459.
21. Woods, D. R., A. N. Hrymak, R. R. Marshall, P. E. Wood, C. M. Crowe, T. W. Hoffman, J. D. Wright, P. A. Taylor, K. A. Woodhouse, and C. G. Bouchard. 1997. Developing problem solving skills: The McMaster problem solving program. *Journal of Engineering Education,* 86 (2): 75-91.