2006-390: AN ARCHITECTURE FOR VIRTUAL LABORATORY EXPERIMENTATION

EL-Sayed Aziz, Stevens Institute of Technology

Dr. El-Sayed Aziz holds a faculty position in the Mechanical Engineering Department at Mansoura University, Egypt. Currently, he is working as research scientist at Stevens Institute of Technology in Hoboken, New Jersey, USA. He received B.S. and M.S. degrees in Mechanical Engineering from Mansoura University, Egypt, in 1991 and a Ph.D. in Mechanical Engineering from Stevens Institute of Technology in 2003. His research interests include knowledge-based engineering systems; computer-integrated design and manufacturing; Finite Element Analysis; software development and applications; as well as remote and virtual laboratories.

Sven Esche, Stevens Institute of Technology

Dr. Sven K. Esche is currently holding a position as Associate Professor of Mechanical Engineering at Stevens Institute of Technology in Hoboken, New Jersey, USA. In 1989, he received an undergraduate degree in Applied Mechanics from Chemnitz University of Technology (Germany). After working for three years at Mercedes Benz AG in Stuttgart (Germany), he obtained M.S. and Ph.D. degrees in Mechanical Engineering from The Ohio State University in Columbus, Ohio, USA in 1994 and 1997, respectively. His current research interests include multi-scale modeling of thermo-mechanical processing of metals, integrated product and process design under conditions of uncertainty and risk as well as remote sensing and control of distributed devices with special focus on remote laboratories.

Constantin Chassapis, Stevens Institute of Technology

Dr. Constantin Chassapis is Professor and the Director of the Mechanical Engineering Department, Professor, at Stevens Institute of Technology. His research interests are in knowledge-based engineering systems; computer-aided design and manufacturing; structure-property modeling and characterization of polymers and polymer composites. He has received the best paper award from SPE’s Injection Molding Division, the distinguished Assistant Professor Award at Stevens, an Honorary Master’s Degree from Stevens Institute of technology, and the Tau Beta Pi Pi Academic Excellence Award. He has been an active member in ASME, and SPE.

An Architecture for Virtual Laboratory Experimentation

Abstract

This paper presents a software architecture for the rapid development of virtual laboratories that support a flexible online collaborative learning environment. The experimentation system integrates Java, Python and Virtual Reality Modeling Language (VRML) to provide multiple users with virtual laboratories over the Internet.

The system architecture is modular and can be easily extended to implement different experiments. Most of the system components are implemented as Java applets that can be run on commonly available Web browsers in an interactive way, thus minimizing the network overhead. The proposed approach and environment have been implemented and are currently tested in different experiments such as a mechanical vibration system, a muffler system, a liquid level control system and beam deflection system. This environment provides the students with the possibility to learn in a flexible way, i.e. they can follow different learning modalities to perform experiments from remote locations using the Web browser. This paper will summarize specific outcomes and provide sample screenshots at key stages of a representative experiment – the classical strength-of-materials problem of determining the deflections and stress concentrations of a cantilever beam of linear elastic material with stress raisers.

Index Terms: Laboratory education; Internet; on-line experiment; Web-based laboratory; virtual laboratory; virtual experiment; virtual reality; VRML.

Introduction

The fast development of computer systems along with the spreading Internet connectivity have opened the door for flexible delivery of engineering education and for providing students with a very rich source of educational material. Many educational institutions have implemented virtual and remote laboratories as an Internet-based learning environment to support flexible engineering education curricula. Gillet et al.[1,2,3,4] described the collaboration between the Swiss Federal Institute of Technology in Lausanne, Switzerland, and The University of Florida in Gainesville, USA, to develop and share Web-based experimentation resources. Their environment integrates all the components necessary to carry out hands-on practice in a flexible learning context. It has the capability to reach more students by enabling them to access the experimental devices from remote locations at anytime via the Internet. At Purdue University, a virtual laboratory called SoftLab has been developed to provide an environment for both physical experiments and numerical simulations[5]. Web-based experimentation is becoming an increasingly attractive way in the deployment of e-Learning solutions and the modality of choice for enhancing the laboratory experience. It can include virtual (i.e. simulation-based) and remote (i.e. hardware-based) laboratory resources.

Simulation-based virtual laboratories have become the leading solution to provide for the needs of the e-Learning community. They represent a valuable option in academic laboratories due to their advantages: inexpensive operation and no time and physical restrictions. They also provide a safe learning environment for experimentation with dangerous equipment[6]. If the virtual reality simulation is very detailed, it can be used as a good substitute for realistic hands-on experimentation and can provide educationally valuable features not available in hardware-based experiments. Students can perform experiments on simulated systems by means of special software provided by a server through the Web browser without downloading the source code. For example, a Virtual Engineering/Science Laboratory at Johns Hopkins University makes it possible to simulate engineering and science laboratory projects on a remote computer[7]. This environment mainly focuses on experiments for demonstrating theoretical concepts and runs without veritable experiment instruments. It provides interactive, Web-based experiences aimed at increasing the students' understanding of the general principles involved in the experiments.

Recently, many remote laboratories have been developed, but the majority of them use text based interfaces for entering input parameters. Therefore, they could be considered to be an inadequate replacement for real laboratories[8]. The remote access to the actual laboratory equipment through the Internet suffers from a network overhead that reduces the interaction between the student and the system[9]. On the other hand, the simulation approach, taken to the client side, does not cause this problem. The remote laboratory experiment may not comprise all the features imaginable, but some of those can be demonstrated with virtual laboratories. The incorporation of new elements or leading the experiment to extreme situations could be infeasible using the real equipment. For example, students use a real beam setup (see Figure 1), apply a sequence of given forces at the free edge of the beam through a graphical user interface (GUI), and submit the request to a Web server[10,11,12]. They then receive the experimental results delivered back to the client side.

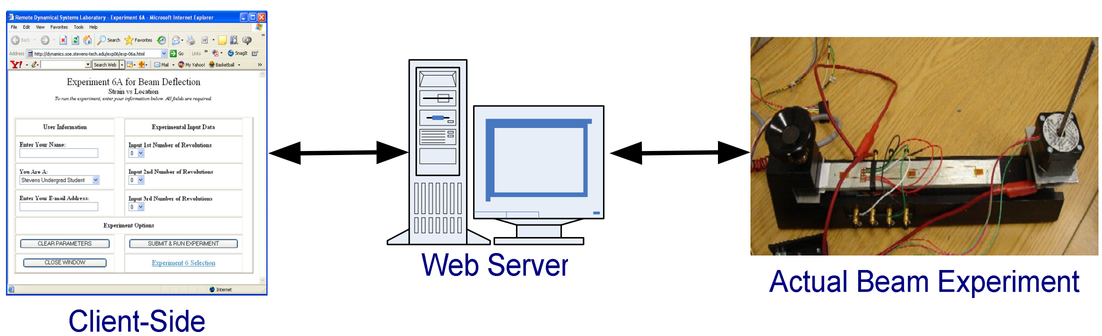


Figure 1: Remote Beam Experiment

With the development of the virtual laboratories described in this paper, the students furthermore have the ability to change the beam dimensions, the beam material, and the location along the beam axis or laterally besides the hole where the strain is to be determined. These parameters were included into the simulation model, and their effects can thus be demonstrated with the virtual laboratory.

Architecture of Virtual Laboratory

A virtual laboratory is an interactive software-based environment for conducting simulated experiments. Such a virtual laboratory is currently being developed for the laboratory courses of the Mechanical Engineering Department at Stevens Institute of Technology (SIT). It mainly focuses on experiments to demonstrate theoretical concepts. The simulation environment has been designed to convey a strong feeling of immersion, as if the students were performing a real-world experiment. Each experiment is implemented in two different versions, as shown in Figure 2, one based on the stand-alone application program and the other based on a Web server and Web browser architecture. Both of them are facilitated by accessing a multimedia assistant.

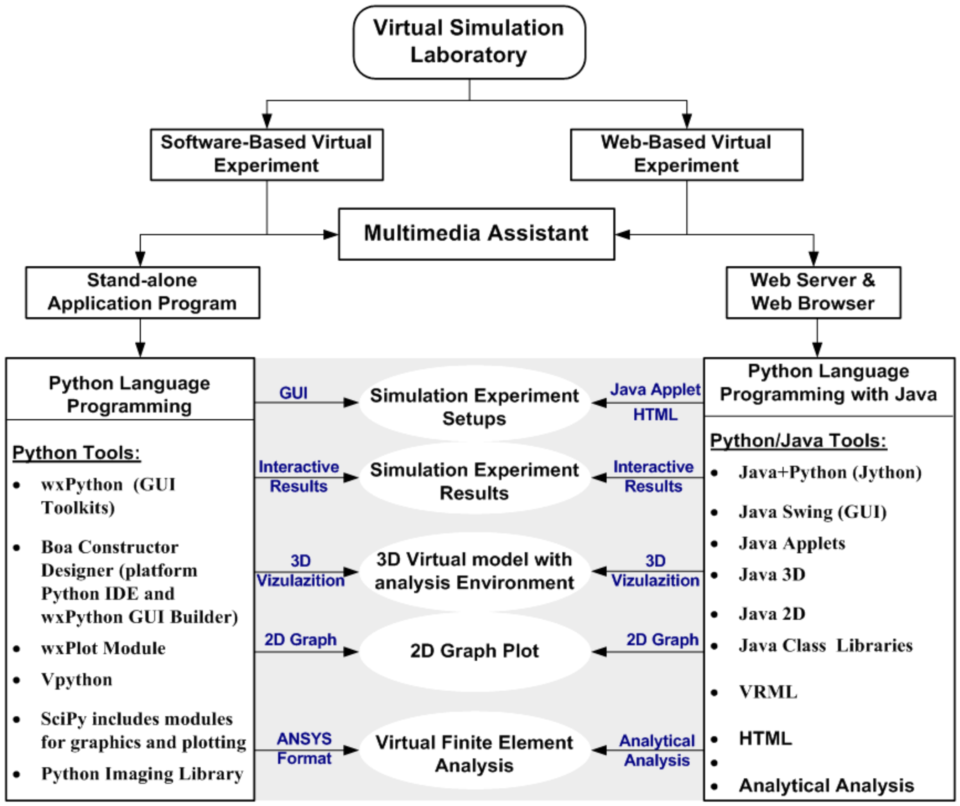


Figure 2: Virtual Laboratory Structure

The stand-alone application programming is developed using the Python Programming Language[13] and wxPython[14] as a cross-platform toolkit. The main features of the 3-D virtual model are developed by using VPython[15] which includes the Python Programming Language, the IDLE interactive development environment and “Numeric”, a Python module for fast processing of arrays. This environment provides a full-featured virtual-reality user interface that allows the experimental configuration and parameters to be controlled interactively by the students. Moreover, it is enabled to generate an appropriate input file to the FEM solver for analyzing the beam experimental setup using a 3-D FEM model. This simulation software must be executed on each student’s remote computer after downloading the compiled code.

One of the emerging uses of the Internet in engineering education is to make laboratory facilities available to a wider student community. In this way, several students are able to perform the simulations simultaneously through the Internet from anywhere at any time. Java applets have almost the same capabilities as the stand-alone application programs, including operating interactively with the user and interacting with resources on the server. Therefore, in the second development, Java applets embedded in HTML files are used to construct a user-friendly interface for the simulations. They are easy to access, easy to update and platform-independent. The Java Applets are written using Jython[16]. Jython is an implementation of the Python

Programming Language which is written in pure Java. It combines the advantages of Python and the Java virtual machine[17] and library (such as Swing, Java Cryptography, Java API, etc.) and serves as a handy complement to the Java platform. Specifically, development times with Jython can be shorter than with Java, owing to the generally shorter code and the lack of a compile phase.

VRML is a modeling language that can be used to build virtual 3-D scenes and script nodes on the World Wide Web. It works even with low-bandwidth Internet connections because of the small amount of transmitted data[18]. Java and VRML are tools with tremendous potential for creating 2-D and 3-D visualization over the Internet. For communicating and massage passing between Java and VRML, all VRML objects are made in the Java applet, and then the whole virtual experiment model is sent as text to the VRML environment. VRML is platform-independent, and it runs usually as a plug-in to standard Web browsers. A VRML plug-in is needed on the client side to interpret the 3-D scenes. The plug-in is available for most systems and can be downloaded from the Internet for free.

Integrating Java/Python for the interface and physical based modeling, VRML for the 3-D scripting and visualization and HTML provides interactive 3-D graphics plus complete programming capabilities and network access. This is a powerful combination to support the building of web-based virtual experimentation systems.

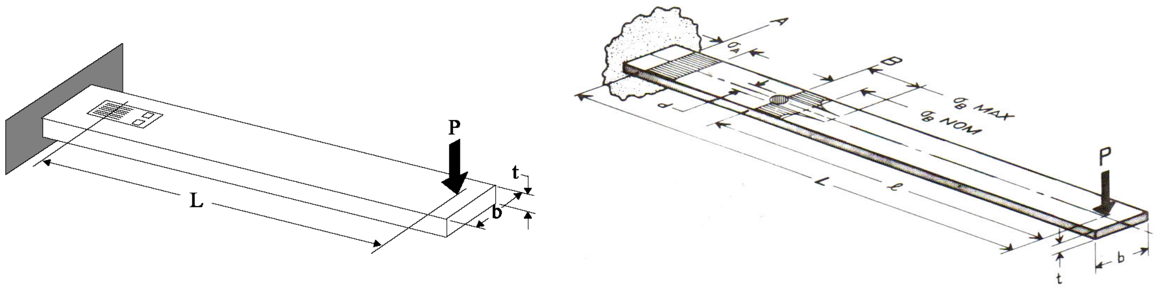
The aim of the virtual simulation laboratory is to provide access to experimental simulations via the Internet to students. The Web page at http://dynamics.soe.stevens-tech.edu/website/ hosts the various simulations that are currently being implemented and tested in our laboratory, such as:

- Mechanical vibration system
- Muffler system
- Liquid level system
- Cantilever beam deflection system

The simulation model of the cantilever beam experiment is used to describe in detail the operations of the features of the virtual laboratory discussed above.

Cantilever Beam Experiments

The cantilever beam is a widely used structural element[19], for example in airplane wings, supports for overhanging roofs, the front spindles of automobiles, etc. A cantilever is commonly defined as a beam which is built-in and supported at only one end, and loaded by one or more point loads or distributed loads acting perpendicular to the beam axis. This experiment is to verify the relationship between the bending-moment, and stress-strain distributions along the length of a cantilever beam. The objective of this experiment is to teach all undergraduate engineering majors at SIT how to use strain gages for the measurement of strain (the axial elongation of the beam). A brief description of the implementation and experimental results for the beam deflection system is included here. Under the beam deflection system, there are two experiments as shown in Figure 3:



Beam without Hole Beam with Hole

Figure 3: Schematic of Cantilever Beam System

a) Beam without hole: This experiment studies the uniform cantilever beam rigidly clamped at its fixed end and deflected by a single point load on the beam centerline near the free end. Three strain gages are installed at equal intervals along the axis of the beam as shown in Figure 4. As the students learn in class, the stretching of an electrical conductor increases its resistance. The gages are designed to take advantage of this effect to measure the strain. The purpose of this experiment is to determine the shear force and the load from the strain measurements, to verify the linearity of the strain along the beam axis, and to confirm the shear force and moment relationships by comparing two different methods for determining the stresses.

b) Beam with hole: The purpose of this experiment is to demonstrate the existence of stress and strain concentrations in the vicinity of a geometric discontinuity (here a circular hole on the beam centerline) in a cantilever beam, and to obtain an approximate measure of the elastic stress concentration factor, (K). In measuring the stress concentrations, three strain gages are installed for measuring the strain field near the hole, and the fourth gage, which is located close to the fixed end and in the center of the beam, varies linearly along the length.

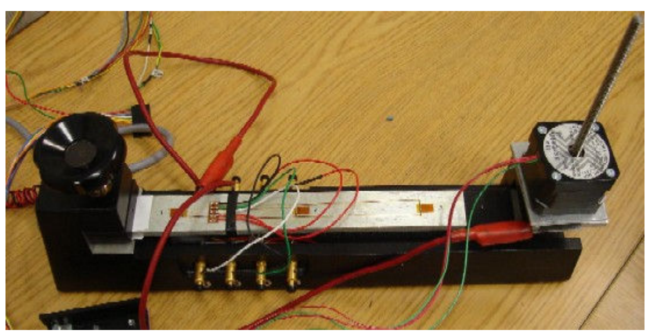


Figure 4: Experimental Setup of Beam without Hole

Cantilever Beam Virtual Experiment Software

The experiment software for the cantilever beam is a stand-alone application program that was developed with the Python Programming Language. It provides the best performance from an interaction point of view. It expands the scope of the experimentation beyond the limits of the hardware-based experiment, allowing the students to select the beam type and the beam material, and enables them to change the beam parameters (length, width and height) and to apply a greater range of transverse loads. A sample snapshot of the software design for the beam experiment is shown in Figure 5. The experiment is used to examine the stress and strain distributions in an elastic cantilever beam under various transverse loads applied at the free end as well as the stress concentrations around geometric discontinuities such as holes.

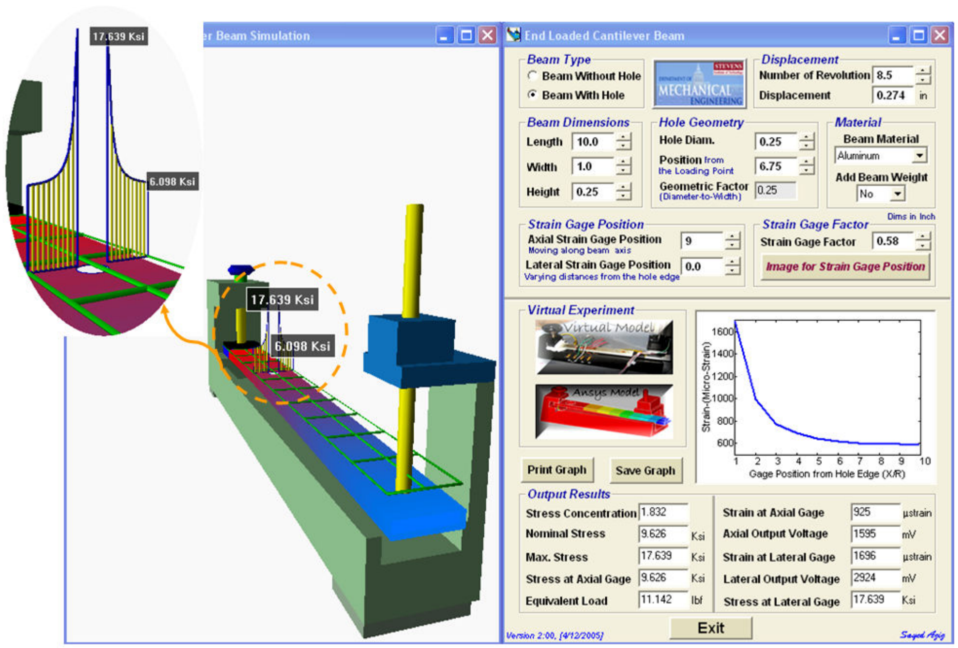


Figure 5: Cantilever Beam Virtual Experiment Software

The software design can be broken down into the following three major groups:

1. Experiment simulation setups with the following main features:

    - The beam type is selected by clicking on the appropriate radio button for a beam without hole or a beam with hole.
    - The geometric parameters of the beam such as length, width and height are defined by a spinning button or by typing the value into the text field area for each parameter.
    - A displacement is applied at the free end of the beam by spinning the "number of revolution" button up or down.
    - If the radio button of the beam with hole is selected, the hole dimensions are activated for specifying the hole diameter and the hole position along the axis of the beam.
    - The beam material is selected, and optionally the beam weight is added to the applied load.
    - The axial and lateral strain gage positions are specified by moving them along the axis of the beam or transversely from the edge of the hole to the edge of the beam, respectively.
    - The strain gage factor supplied by the manufacturer is defined by spinning the corresponding button or by typing the value into the corresponding text area.

- For help, one clicks on the image of the "strain gage position" button to see how to define the strain gage position in the axial and lateral directions.

2. Virtual design environment with 2-D and 3-D visualization:

   - By clicking on the "virtual model" button, the program automatically invokes a 3-D model with analysis environment. Then, the beam deflects, and for visual clarity the color contour of the beam surface changes to indicate the magnitude of the normal stress.
   - For plotting data with zoom, labels, and automatic axis scaling, the simulation interactively generates a 2-D graphical output that visualizes the linearity of the axial strain gage along the beam axis.
   - By selecting the "beam with hole" radio button, the simulation generates a 2-D graph for the strain distribution around the hole, and the 3-D virtual model includes a 3-D stress distribution graph around the hole edge.
   - The student is enabled to move around the beam experiment model and to view it from any angle.
   - The stresses and deflections for any magnitude of applied load are easy and quick to compute based on the concepts of strength of materials.

3. Simultaneously outputting results and measuring simulation data:

   - Students can determine the shear force and the load from the strain measurements.
   - Students can display the stress and strain concentrations in the vicinity of a geometric discontinuity (such as a hole) in a cantilever beam.
   - Students can record the output voltages of the longitudinal and transverse strain gages and compare the simulation with the physical experiment results.
   - Students can visualize, tabulate and graph data, as well as process and present results.
   - Students can print out the 2-D graph for the strain distribution and save the graph as a file in five different file formats (.bmp, .xbm, .xpm, .png and jpg).

Virtual Finite Element Analysis

In order to achieve a more complete understanding of the influence of the real working conditions on a laboratory experiment on the bending of a cantilever beam and to obtain a more accurate estimation of the stress concentration, an FEM model of the 3-D beam experimental setup is employed using ANSYS[20]. The advantage of this approach is to encourage the students to gain additional insight into the fundamental concept of the numerical experimentation and visualization analysis using FEM in mechanics of materials.

The FEM model is automatically generated, either within the Java Applet executing on the Web browser or within the stand-alone system running on the local computer, and the data are written into an appropriate file format for the ANSYS FEA solver which contains the geometric parameters of the beam and the cantilever flexure frame experiment as shown in Figure 6.

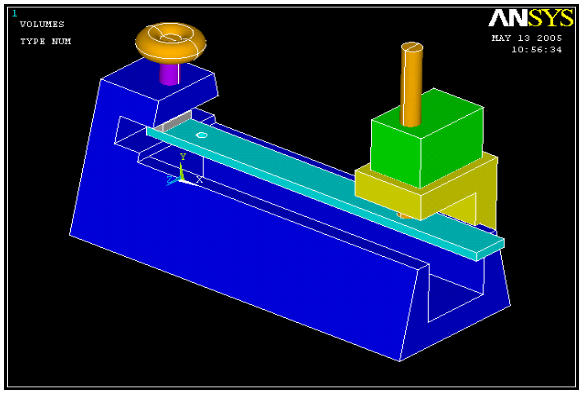


Figure 6: FEM Beam Experiment Model

Once the material properties, such as the Young's Modulus and Poisson's ratio, are picked by the student through the material selection from the main panel of the cantilever beam virtual experiment software (see Figure 5), they are automatically transferred to the FEM solver. The generation of the boundary condition data and the mapping of these data to the model are handled through the workstation-based system output file, which serves as an input to the FEM solver. The concentrated transverse load applied at the free end of the beam and, subsequently, the beam deflection, stress and strain distributions and stress concentration factor for the beam with hole can be determined at the desired position along the axis of the beam, as shown in Figure 7. This procedure is totally automated and can be made transparent to the student.

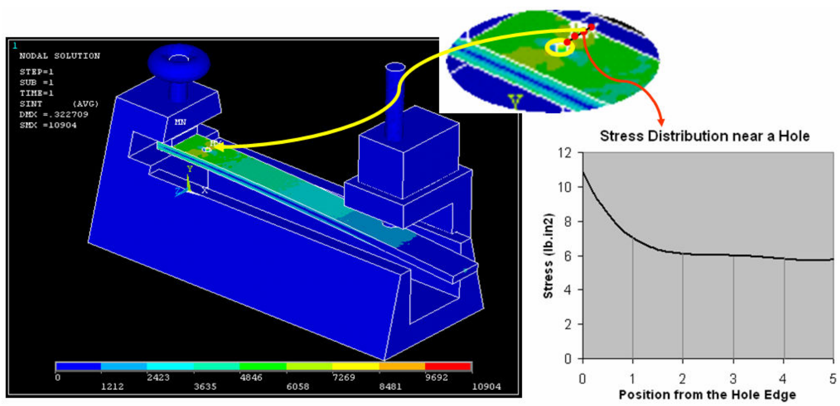


Figure 7: Stress Distribution near a Hole

Web-Based Cantilever Beam Virtual Experiment (Utilizing Web Server and Browser)

The Web-browser based virtual experiment enables both on-campus and distance-learning modes where the users are not required to download any software. When the student selects the beam experiment simulation (which can be found at: http://dynamics.soe.stevens-tech.edu/website), the Java applet is executed on the student's local machine and requires permissions to read and write files on that machine. When the applet is first loaded, the Web browser pops up a dialog window asking for full access permission for the applet to read and write files on the client machine, as shown in Figure 8. The corresponding question must be answered with yes. Granting these permissions at most makes the applet have the same capabilities as any other executable (stand-alone application program) that one would run on the local computer.

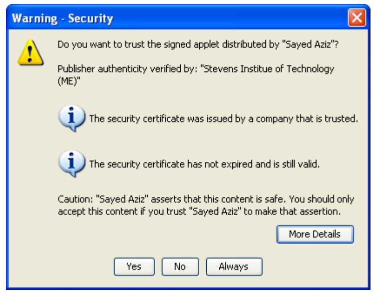


Figure 8: Security Message

Figure 9 shows the layout of the Web page as the students see it in their Web browser. The Web page is composed of two main panels. The left command panel is the Java applet with different options for handling the cantilever beam simulation, such as selecting the beam type, specifying the geometric parameters of the beam, the strain gage and hole positions, the end deflection as well as a pull-down tag to select the beam material (see Figure 5 and the accompanying procedures for more details). The right panel is the 3-D virtual beam experimentation system. During the experimentation phase, changes in parameters and variables are immediately reflected numerically and graphically as a response to the students' actions. The students can simultaneously view the resulting force, stresses and the theoretical deflection curve. Furthermore, the students are also enabled to visualize, tabulate and graph data, thus reducing the time required to process and present results.

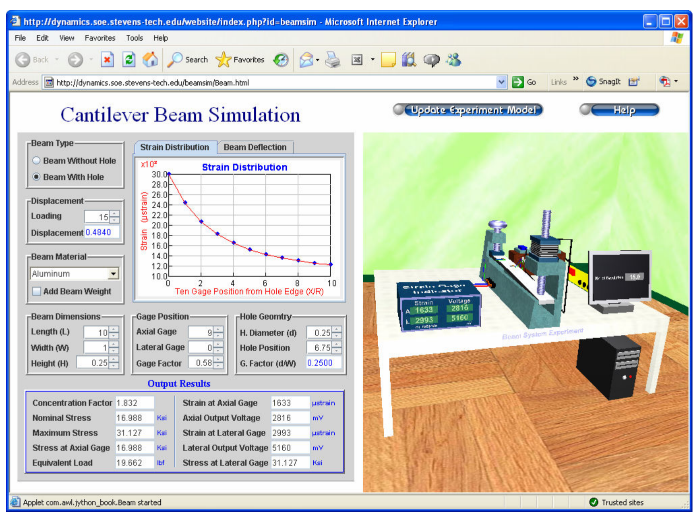


Figure 9: Components of User Interface at Client Site for Cantilever Beam Experiment

The Java Applet provides interactivity between the users and the VRML worlds and makes VRML fully functional and portable. All VRML objects of the beam experiment are generated within the Java applet, and then the whole virtual model is interactively sent as text to the VRML environment on the student's machine. As shown in Figure 10, the simulation model was developed to convey a strong feeling of immersion. The students can navigate the experimental set-up, examine or rotate inside the virtual environment using the standard VRML navigation features. This allows them to move around the beam experiment model and view it from any angle. Students can zoom in at the strain gage positions in the axial and lateral directions to see how the strain gages attach to the upper surface of the beam.

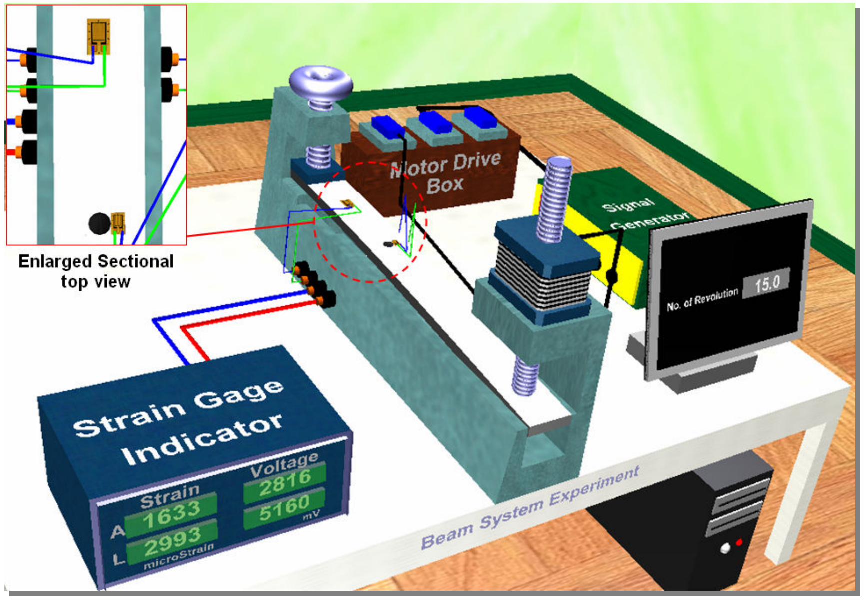


Figure 10: Virtual Beam Experiment

Figure 10 shows a strain gage bonded to the top surface of the cantilever beam and connected to the strain gage indicator through a wire connection to measure the axial deformation of the beam (the deformation along the length of the beam) when a transverse load is applied to the end of the beam. The load is applied in the form of a number of revolutions of a power screw as shown in the screen, which is transferred to a signal generator, to the motor-drive box to operate the stepper motor to turn the power screw to bend the beam with the appropriate displacement. Based on the mathematical model developed to simulate the real beam experiment, the output strain and voltage values appear on the screen of the strain gage indicator.

Conclusion

This paper presents the development of a virtual laboratory which enhances the students' understanding of physical concepts by providing a high degree of immersion within rich interactive learning environments. The experimentation system integrates dynamic and interactive simulations in stand-alone and Web-based environments. During the experimentation phase, changes in parameters and variables are immediately reflected in the graphical user interface. Thus, the students can visualize in real time how the model behavior changes according to the values of the interactive variables. All software developed for the virtual laboratory was implemented in the programming languages VRML, Java and Python. Therefore, it can be easily adapted to different platforms.

As an example for the virtual laboratory environment presented, the classical strength-of-materials problem of determining the deflections and stress concentrations of a cantilever beam made of linear elastic material with stress raisers is described. The students can interactively visualize, tabulate and graph data, and process and present results to explore the structural behavior.

The future plans for this project include the evaluation and revision of features through student assessment and the addition of other experiments and simulations.

References

[1] Gillet, D., Latchman, H. A., and Salzmann, C., (2001), "Hands-on Laboratory Experiments in Flexible and Distance Learning", Journal of Engineering Education, April, pp. 187-191.

[2] Gillet, D., Geoffroy, F., Zeramdini, K., Nguyen, A. V., Rekik, Y., and Piguet, Y., (2003), "The Cockpit: An Effective Metaphor for Web-based Experimentation in Engineering Education", Int. J. Engng. Ed., Vol. 19, No. 3, pp. 389-397.

[3] Nguyen, A. V., Gillet, D., Rekik, Y. and Sire, S., (2004), "Sustaining the Continuity of Interaction in Web-based Experimentation for Engineering Education", Proceedings of CALIE, Grenoble, France.

[4] Nguyen, A. V., and Gillet, D., (2003), "Web-based Collaboration in Flexible Engineering Education", SIGGROUP Bulletin, December 2003, Vol. 24, No. 3, pp. 117-128.

[5] Catlin, A. C., Gaitatzes, M. G., Houstis, E. N., Ma, Z., Wang, N.-H., and Weerawarana, S., "The SoftLab Experience: Building Virtual Laboratories for Computational Science", Online document available at: http://www.cs.purdue.edu/research/cse/softlab/softlab-vlabs/softlab-framework/softlab_report/report.html.

[6] Eriksson, H., Linkopings University, Sweden, "Control the Nuclear Power Plant (Demonstration)" Online document at: http://www.ida.liu.se/~her/npp/demo.html.

[7] A Virtual Engineering/Science Laboratory (http://www.jhu.edu/~virtlab/virtlab.html).

[8] Nedic, Z., Machotka, J. and Nafalski, A., (2003), "Remote Laboratories versus Virtual and Real Laboratories", 33rd ASEE/IEEE Frontiers in Education Conference, November 5-8, 2003, Boulder, CO, USA.

[9] Anido, L., Llamas, M., and Fernández, M. J., (2001), "Internet-based Learning by Doing", IEEE Transactions on Education, Vol. 44, No. 2.

[10] Esche, S. K., Chassapis, C., Nazalewicz, J. W. and Hromin, D. J., (2003), "An Architecture for Multi-User Remote Laboratories", World Transaction on Engineering and Technology Education, Vol. 2, No. 1.

[11] Esche, S. K., (2006), "On the Integration of Remote Experimentation into Undergraduate Laboratories-Pedagogical Approach", International Journal of Instructional Media, Vol. 33, No. 3.

[12] Esche, S. K., (2005), "On the Integration of Remote Experimentation into Undergraduate Laboratories-Technical Approach", International Journal of Instructional Media, Vol. 32, No. 4.

[13] Python Program Home Page at: http://www.python.org/

[14] The home of wxPython at: http://www.wxpython.org/

[15] The home of VPython at: http://vpython.org/

[16] The Jython homepage at: http://www.jython.org/

[17] The Java homepage at: http://java.sun.com/

[18] Masár, I., Röhrig, C., Bischoff, A., Gerke, M. And Hoyer, A., "A Virtual Laboratory for an Inverted Pendulum and Crane Control", available at: http://prt.fernuni-hagen.de/~bischoff/research/pdf/Masar04a_TA04.pdf

[19] Education Division, Measurements Groups Inc., "Experiments in Mechanics, Strain Gage Series: (1) E-104 Stress and Concentration and (2) E-105 Cantilever Flexure", P.O. Box 27777, Raleigh, North Carolina 27611.

[20] ANSYS, Inc. homepage at: http://www.ansys.com/