UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

IMPLEMENTATION OF THREE-DIMENSIONAL VISUALIZATION IN INTERACTIVE WEB-BASED ENVIRONMENT FOR ENGINEERING EDUCATION AND TECHNICAL TRAINING

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

CHATURAPORN NISAGORNSEN

Norman, Oklahoma

2003

IMPLEMENTATION OF THREE-DIMENSIONAL VISUALIZATION IN INTERACTIVE WEB-BASED ENVIRONMENT FOR ENGINEERING EDUCATION AND TECHNICAL TRAINING

A Dissertation APPROVED FOR THE

SCHOOL OF AEROSPACE AND MECHANICAL ENGINEERING

ACKNOWLEDGEMENTS

First of all, I would like to express my utmost appreciation to my advisor, Dr. Kurt Gramoll, for his guidance, mentoring, support, persistence, and assistantship. Without which, my dissertation and research work would not have been made possible. His charismatic leadership, excellent management and technical skills never cease to astonish me, and I am very honored to have the opportunity to work under his tutelage. I also wish to express my sincere gratitude to Dr. Thomas Landers, Dr. Simin Pulat, Dr. Kuang-Hua Chang, and Dr. Zahed Siddique for serving on my advisory committee and for their suggestions and cooperation throughout my research and study. I especially thank Dr. Siddique for his assistance, encouragement, and advice on the research project, which is also a part of this dissertation.

I am grateful to my colleagues at the Engineering Media Lab for their support in making my research work an enjoyable experience. I offer special thanks to my friends for making my graduate study a pleasurable journey. I would also like to extend my sincere gratitude to Jennifer Connelly and the staff of the University of Oklahoma Writing Center for her patience and assistance in helping me during the writing process.

Above of all, I offer heartfelt expression of gratitude is to my mother, Chantana Somwang, and my sisters for all their endless love, support, patience, understanding, and encouragement that helped me throughout the course of my education. Also, I thank them for giving me this wonderful opportunity and experience for a better education and for providing me with financial needs. Without these from the other side of the globe, this work would have been extremely difficult or likely impossible. My dedication of this dissertation goes out to all of you, and particularly, to my delightful mother.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The rapid advancement of the Internet and web technology has provided tremendous possibilities for various web-based applications such as education and training. In particular, engineering and technical topics are among the most challenging topics developed and brought to life in a web-based environment. In web-based environments, three-dimensional visualization has played a major role as one of the fundamental elements for engineering and technical web-based environments. However, educating and training people on a technical subject using the web-based environment still has some key obstacles that have been a major concern in incorporating three-dimensional visualization for more effectiveness and convenience. Fortunately, in the past few years the advancement of technologies that can be used on the Internet and the web is the main factor that makes convenient the tremendous possibility of having an effective web-based education and training for engineering and technical areas.

The main focus of this research is to develop, experiment with, and demonstrate the concept of using the latest potential technology with web-based three-dimensional graphics supportive capability that can bring more efficiency and convenience to the world of web-based engineering education and technical training. The unique feature and technique that is used in this research to make a significant improvement in web-based environments for training in engineering and technology is the introduction of Shockwave 3D technology. To demonstrate the effectiveness of the Shockwave 3D technology, illustration environments have been developed. In order to reach the entire objectives of this research, two environments are presented. The first environment, which

is called the CSD System, concentrates on technical training and the other one, namely the 3D Dynamics System, focuses on engineering education.

The first system, the CSD System is not only an illustration environment, but also an actual system that has been built for Tinker Air Force Base (AFB). The main purpose of Tinker AFB having this web-based environment developed is to provide a flexible and effective training system that is accessible as a resourceful, stimulating, attractive, and challenging training method to serve a number of personnel simultaneously, while catching the attention of personnel, which is the utmost concern. Tinker AFB intends to use this interactive web-based training system to guide and instruct its base personnel to perform the specific *Aircraft Part Maintenance and Assembly Process* procedures properly. The other demonstration, the 3D Dynamics System is an interactive web-based educational environment developed to assist the student in learning basic concepts in *Engineering Dynamics* concentrating on the topic of *Three-dimensional Rigid Body Motion.*

Moreover, in order to clearly understand the development of these two innovative systems, the details of the web-based training system developed for Tinker AFB, namely the EM/C135 System, are discussed first. Since this system is considered the foundation system for the demonstration environments, especially, for the 3D Dynamics System, which is an extended section of the EM/C135 System. This system is the development of an interactive web-based training system that incorporates the capabilities of different Internet technologies to provide a cohesive, independent and self-contained environment that can aid in learning for both military and civilian personnel at Tinker AFB through inclusion of training material for environmental engineering and aircraft maintenance instruction. The purpose of the Environmental Engineering project is to train and guide the Air Force Base personnel, including both technicians and engineers, to comply with the standard requirements of various NESHAP (National Emission Standard for Hazardous Air Pollutants) and environmental regulations. The C-135 Aircraft Maintenance Instruction project gives detailed instructions on the correct maintenance procedures for the pressurization on the KC-135 series aircraft.

CHAPTER 1: INTRODUCTION

The Internet is changing the method of education because it can provide new and creative methods for presenting course material interactively. On November 16, 1999, John Chambers [1], CEO of Cisco, delivered a keynote address to the Comdex audience in Las Vegas. In his address, Chambers stressed taking advantage of e-learning as a "second wave" of the Internet. He also emphasized that through e-learning, employees will have life-long learning opportunities with material available 24 hours a day without constraints in location and schedule.

An advantage of web-based education is the flexibility of letting students adapt to their own learning pace and different ways of understanding concepts. A feature that allows immediate feedback and develops an interactive and intuitive learning environment can deepen the student's ability to understand the subject. The facility to enroll in courses offered from other institutions and the capability to deliver material across different computer platforms made the Internet an increasingly popular medium for distance learning [1, 2].

Another advantage is that web-based education is an excellent alternative solution to solve the problems that occur when learner populations increase, and organizations face the challenge of maintaining high standards for effective education and training. This convenient and inexpensive approach is capable of complementing traditional education and training. This is particularly true in the resource-intensive engineering and technical fields, where engineers and technicians must always continue learning in order to master the latest technologies.

Interactivity in a web-based environment has been adopted widely to effectively deliver the material to the student, such as engineering design and analysis [3, 4, 5, 6] and understanding machine operation and manufacturing process [7]. Numerous technologies such as audio, video, animations, simulations and text with standard graphics have been used to support engineering education. To be successful in demonstrating engineering concepts, especially in mechanical engineering, the best way to present information to the user is to use three-dimensional models [8]. As concepts of virtual reality become possible, the use of three-dimensional elements can simplify the learning process. Furthermore, integrating the concept of three-dimensional visualization with interactivity to create simulations and virtual environments creates an exploratory world in which students can learn through experimentation and research.

Obstacles of the Existing Web-based Engineering Education

In order to better understand engineering and technical concepts, an active learning process needs to be stimulated. By incorporating catalysts such as three-dimensional graphics, simulations, and animations, an active learning environment can be developed. Although there are images of three-dimensional graphics, their two-dimensional format does not imitate the actual environment because objects are three-dimensional. To help bring this idea to reality on the web, Virtual Reality Modeling Language (VRML) and Java 3D technologies were developed [9, 10]. Presently, the three-dimensional visualization that is integrated in most of the existing interactive web-based environments is developed using either VRML or Java 3D technologies. These two technologies are most widely used to implement a three-dimensional world on the web.

The probability of VRML being accepted as a good solution is possible because of the acknowledgement of it as a global web-standard for three-dimensional visualization. VRML is a suitable environment to navigate in the three-dimensional world because it provides extensive support to interactivity, animation, modification, and hyper linking. Furthermore, VRML uses easy-to-understand text to describe the three-dimensional model, allowing the VRML file to be edited in any text editor. Many organizations around the world have done research in developing VRML models on the web in a variety of fields, including engineering.

The obstacle for VRML is finding a way to view VRML models conveniently as a part of the education environment. A specific plug-in or player for VRML, which is not commonly used, is required. Moreover, VRML is outstanding at presenting static three-dimensional objects, but needs the help of other programming languages to create dynamic objects and to conduct complex computations such as JavaScript, Perl (Practical Extraction and Report Language), and ASP (Active Server Pages) script. To create more complex and dynamically generated geometric objects, other languages with full capabilities, such as Java and Java 3D, are required.

Java is a general-purpose programming language that is designed for the Internet. The key characteristics of Java are: simple, object-oriented, multi-threaded, robust, high-performance, secure, platform-independent, portable, and it is a network-supporting environment. To design a geometric modeling application using Java, a three-dimensional application programming interface such as Java 3D is needed. Java 3D is a standard extension of Java used to provide a three-dimensional related library in order to design three-dimensional applications. Java 3D works smoothly only in a Java environment and uses Java as the user Interface.

Similar to the obstacle faced using VRML, three-dimensional visualization developed using Java and Java 3D require their own plug-ins as well. The problems of these plug-ins are not only getting them but also using them. Plug-ins for Java and Java 3D are rarely used due to their availability, they have problems with big file sizes, firewalls block, security issues, and people are reluctant to install them due to viruses. Additionally, although simplicity is a significant characteristic of Java, to understand, use, and implement this technology is still more complicated than to do so with VRML. In order to develop an environment where three-dimensional visualization can be made a part of the learning environment, use of other technology and different techniques needs to be investigated. This technology should be able to overcome the obstacles of present technologies.

Solution

In the past, Shockwave technology was used to develop applications with complicated static and dynamic graphics as a part of two-dimensional learning environments. As good performance results with two-dimensional visualization have been obtained, the constraints of using an extra plug-in to incorporate a three-dimensional world as a part of the learning environment can be eliminated by using the latest version of Director, namely Director 3D, that is used to develop the Shockwave. This latest capability requires only the Shockwave plug-in to implement a web-based three-dimensional learning environment. The automatic downloading feature of Shockwave assists in minimizing users' trouble with finding appropriate and widely installed plug-ins that are widely accepted as secure and get through firewalls. Further, the Shockwave technology has an advantage over the VRML and Java technologies with its inbuilt streaming capabilities.

Research Objectives

The focus of this research is to (1) design an interactive web-based environment for engineering education and technical training, which will illustrate the advantages of using the web to deliver the learning material and the benefits of a learner's ability to interact with the learning environment throughout the learning process. Another objective is to (2) develop an interactive web-based environment by integrating three-dimensional visualization with other Internet technologies to provide a cohesive, independent, and self-contained learning system. Three-dimensional graphics, animations, and simulations are developed to depict the actual situation of the various illustrations, which allows better delivery of detailed instructions on the correct procedures, as well as improved guidance to gain better understanding. This objective will illustrate why engineering concepts can be demonstrated successfully using three-dimensional visualization. However, currently some difficulties exist in implementing three-dimensional visualization, so improved approaches have been investigated. Since no interactive web-based environments exist that incorporate three-dimensional graphics, simulations, and animations to support engineering education and technical training using the three-dimensional graphics supportive capability in the Shockwave 3D technology, (3) developing an infrastructure by effectively collaborating the latest technology of three-dimensional visualization, which is a unique feature presented, is the primary objective of this research.

After the process of design and development of the learning environment using the new technology and technique, the next required process is to verify the efficiency of the overall environment and technology and technique used in the development. Therefore, additional objectives of this research are (4) evaluating the effectiveness of the application to deliver engineering and technical concepts through the Internet and (5) evaluating the efficiency of proposed technology in implementing three-dimensional visualization for interactive web-based environments [1, 2]. Since web-based education and training is still in its infancy, one of the obstacles to web-based education and training is a lack of infrastructure and content-rich applications. In addition, due to the lack of an interactive web-based environment that concentrates on the subject of engineering and technology, this research also aims at (6) providing a foundation for developing content-rich material for engineering education and technical training. Yet another objective of this research is to develop a web-based application that allows instruction in *Aircraft Part Maintenance and Assembly Process* and *Engineering Dynamics: Three-dimensional Rigid Body Motion* courses via the Internet. Further, this research also hopes to achieve the goal of (7) enhancing the efficiency of teaching and learning engineering courses by utilizing various Internet technologies.

The research work includes investigating, designing, developing, implementing, and delivering material using a wide range of multimedia components, such as animation, and simulation with standard text and graphics to deliver relevant information on the subject in order to provide the opportunity for users to explore and learn effectively through an investigative process. Further, these electronic media types are combined to match with many aspects of technical training. To demonstrate the effectiveness of using this latest technology for implementing three-dimensional visualization in an interactive web-based environment for engineering education and technical training, an illustration environment needs to be developed. To achieve the research objectives, two distinguished demonstrations were developed, one concentrates on engineering education and the other one concentrates on technical training. The main purpose of these environments for the research is to serve as prototypes or demonstrations that can provide strong evidences to support the idea of using the Shockwave 3D technology to deliver a better interactive web-based environment as defined by the research goal to (8) identify and demonstrate the implementation of new techniques to provide an improved environment. This results in the development of the CSD System and the 3D Dynamics System as demonstration environments (Note: CSD stands for the *Constant Speed Drive*, which is the name given by Tinker AFB personnel to a demonstration part used in this system. However, the name of this part used in the official Technical Order is *Constant Speed Mechanical Drive Mechanism.*).

Demonstration Environments

The CSD System is the actual system developed for Tinker Air Force Base (AFB) that can be used to demonstrate and support the effectiveness of implementing three-dimensional visualization in an interactive web-based environment for engineering education and technical training using the latest technology, which is the primary focus of this research. Although the CSD System is the actual system, it is built to serve as a demonstration in order for Tinker AFB to explore the potential of initiating a complete web-based training environment in the future. Since it requires assistance from three-dimensional visualization for an efficient presentation, the CSD System is an excellent illustration of the unique three-dimensional graphics supportive capability in Shockwave 3D technology that provides more advantages and conveniences in developing and distributing more effective interactive web-based environments for engineering education and technical training.

The focus of this research work is to provide a framework and infrastructure through this specific illustration module to exhibit how technical engineering subjects such as *Aircraft Part Maintenance and Assembly Process* can be efficiently implemented, educated, and delivered as learning material through the web using three-dimensional visualization as a major component for a presentation that is incorporated with other existing Internet technologies. The fundamental objective of this environment is to help engineers, technicians, and professionals at Tinker AFB understand the subjects presented to be able to perform required operations properly through an internet-based learning environment. These various technologies such as text, graphics, and three-dimensional visualization, including animation and simulation, are integrated to develop and efficiently deliver content-rich learning material. The environment is composed of three major elements including interactivity, three-dimensional visualization, and evaluation. These elements are considered essential elements for this particular web-based environment since they are the minimum elements required for this research to achieve its objectives and to comply with Tinker AFB requirements as well.

The objective of the CSD System is to train and guide Tinker AFB personnel to perform specific *Aircraft Part Maintenance and Assembly Process* procedures properly. This demonstration environment focuses on the *Constant Speed Mechanical Drive (CSMD) Mechanism*, which is composed of a number of subassemblies, including *Left*

Hand Pump and Motor Assembly, *Right Hand Pump and Motor Assembly*, *Governor*, and *Differential*. Each subassembly contains a number of components that varies from at least fifty in *Differential* to at most one hundred and twenty in *Left Hand Pump and Motor Assembly*. However, in this research only the *Left Hand Pump and Motor (LHPM) Assembly* is of interest for an illustration, since the *LHPM Assembly* is the most complicated subassembly in the *CSMD Mechanism*. In addition, this subassembly is considered a main component of the *CSMD Mechanism* as well. The system can be broken down into three major sections: the *Part Breakdown* section presents exploded and unexploded the *LHPM assembly*, the *Assembly Process Tutorial* section provides step-by-step animation for the *LHPM Assembly* procedures, and the *Assembly Process Evaluation* section evaluates the user's understanding of the *LHPM Assembly* procedures through simulation. The step-by-step animation and simulation for the *LHPM Assembly* is developed based on the official Technical Order (TO) document issued by Tinker AFB.

The first section allows a user to access an expandable three-dimensional view of the *LHPM Assembly*. The user can highlight and rotate the selected part at will. The user is then able to select any part in the assembly from an indentured list and the component will be highlighted on the main screen. The selected part will appear solid while the non-selected parts will become more translucent. The user can then select a part from the view port screen and the selected part will be highlighted in the indentured list. This section is used to help users recognize the name of each sub part, what it looks like, and where it is located in the assembly. In the second section, the demonstration of the *LHPM Assembly* is provided to give users the correct procedure to assemble the *LHPM Assembly* step-by-step. Meanwhile, the last section is used to evaluate a user's knowledge of how to perform the *LHPM Assembly* correctly as illustrated in the previous section. The user has to perform exactly the same procedure described and illustrated in the second section to be able to move on to the next step and so on until the completion of the assembly. Figure 1.1 illustrates sample screens of three sections included in the CSD System.

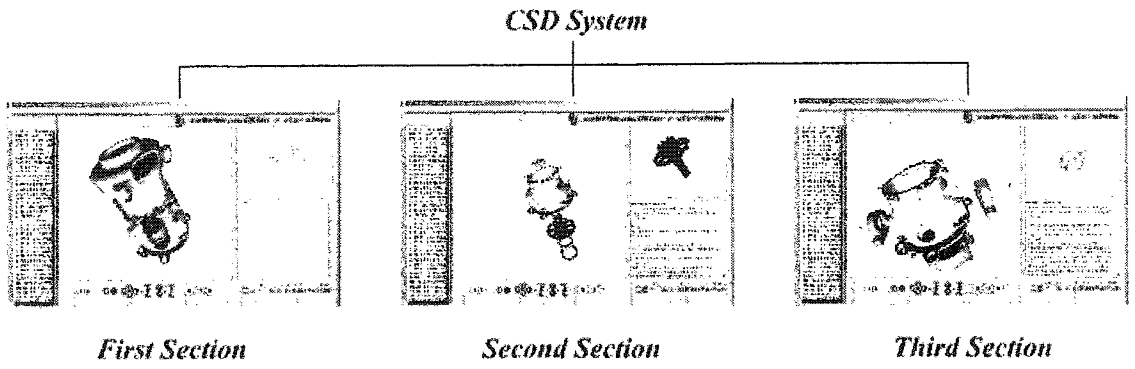


Figure 1.1: Sample screen of three sections included in the CSD System

The 3D Dynamics System is an interactive web-based educational environment developed to assist students in learning basic concepts in *Engineering Dynamics*. This environment concentrates on *Three-dimensional Rigid Body Motion*, covers the same material addressed in a typical, undergraduate *Engineering Dynamics* book, and is designed to serve as a third source in addition to lectures and textbooks from which engineering students can learn about *Engineering Dynamics*. In addition, the system is well suited as a demonstration to accomplish the research objectives because it requires assistance from three-dimensional visualization for an efficient presentation.

In today's academic environment, students sometimes learn theory they cannot transfer to real situations, or have experiences they cannot explain or generalize. With this system the focus has been to integrate computer-based simulations with graphics, audio components, animations, and hypertext, into a self-contained system that will assist students taking a first undergraduate *Engineering Dynamics* course. The goal is to embody the theoretical and discursive aspects of *Engineering Dynamics* with experiential simulations in the same system. The result for the students will expectantly be a more efficient learning process and a deeper understanding of *Engineering Dynamics.*

The system, which is example based, is comprised of various example problems, each of which illustrates a specific concept in *Engineering Dynamics*. Each section is presented in three parts: *Theory*, *Simulation Activity*, and *Problem.* The first and third parts, which incorporate graphics, audio components, animations, and hypertext, introduce a problem to the student, present specific concepts in *Engineering Dynamics* that are required to solve the problem, and apply the concepts to solve the problem. The second part allows the student to experiment with a computer-generated simulation of the problem that is implemented using Shockwave 3D technology. Figure 1.2 illustrates sample screens of three areas included in the 3D Dynamics System.

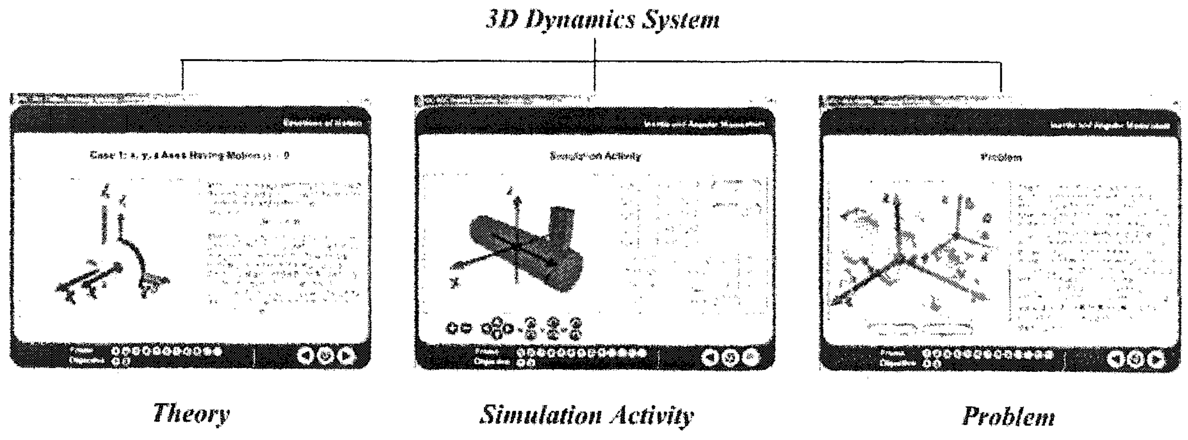


Figure 1.2: Sample screens of three areas included in the 3D Dynamics System

The system includes designing and implementing a new Internet-based learning environment that also features the ability to track and monitor a user's learning progress in understanding *Engineering Dynamics: Three-dimensional Rigid Body Motion.* Information management, such as collecting statistical records of a user's progress, for example the actual time spent and number of times visited on any assigned topic, is to be implemented in order to collect data for the administrative purposes of both the learning system and the sponsoring organization. Designing and delivering material by using a wide range of multimedia components such as animation, sound and simulation with the standard text and graphics is implemented to deliver relevant information on the subject. These electronic media types are integrated to complement with several aspects of technical training. Furthermore, using these technologies provide the opportunity for the user to effectively explore and learn through an investigative process.

The learning system is designed such that both students and administrative staff can use the same interface to access information and is designed to guide, educate, and train the students through use. Several technologies such as text, graphics, animation, sound, and three-dimensional visualization are incorporated for development and delivery of efficiently content-rich learning material. Interactivity is maintained throughout the learning process to make the system participative intensive, so that the concentration of the students is focused on the subject. Implementing and designing simulations to enable students to learn through exploration and investigation along with the learning material is involved. At the end of each learning block, short reviews are also implemented to evaluate and provide feedback to users. Statistical information such as actual time spent and number of times visited on any assigned topic is monitored and recorded in an online database. The information tracked can be used for two purposes. First, it is provided to assist students learn better by making their own assessments. Second, it can be used to evaluate the effectiveness of the system by assessing the statistical information collected from the progress of different users. The significance of tracking such information is that it provides data that relates to the usability of the system over a period of time.

EM/C135 System

Moreover, in order to clearly understand the development of these two demonstrations, the details of the web-based training system developed for Tinker AFB, namely the EM/C135 System, are discussed at first. Since this system is considered the foundation system for the demonstration environments, especially, for the 3D Dynamics System, which is an extended section of the EM/C135 System. This system is the development of an interactive web-based training system that incorporates the capabilities of different Internet technologies to provide a cohesive, independent and self-contained environment that can aid in learning for both military and civilian personnel at Tinker AFB through inclusion of training material for environmental engineering and aircraft maintenance instruction.

This learning system incorporates web-based technology including the ability to track and monitor users. Statistical data on users' progress, such as the actual time spent on each objective, and number of times visited on any assigned topic, is collected for the administrative purposes of the sponsoring organization. A significant advantage of using the web is the availability of a wide range of media that can be implemented effectively for presentation. Information delivery is done by integrating a variety of multimedia components, such as video, animations, sounds, and simulations with the standard text and graphics. Additionally, users' are provided with the opportunity to explore and learn effectively through an investigative process using these technologies.

The purpose of the Environmental Engineering project is to train and guide base personnel, including both technicians and engineers, to comply with the standard requirements of various NESHAP (National Emission Standard for Hazardous Air Pollutants) and environmental regulations. This project is categorized into modules, wherein each module is dedicated to a single regulation applied at Tinker AFB. Each module consists of several individual lessons, and each lesson is further subdivided into objectives, which are the basic learning blocks of the system. At the end of each objective there are activities designed to evaluate users' understanding of the material. The project structure is organized as illustrated in Figure 1.3.

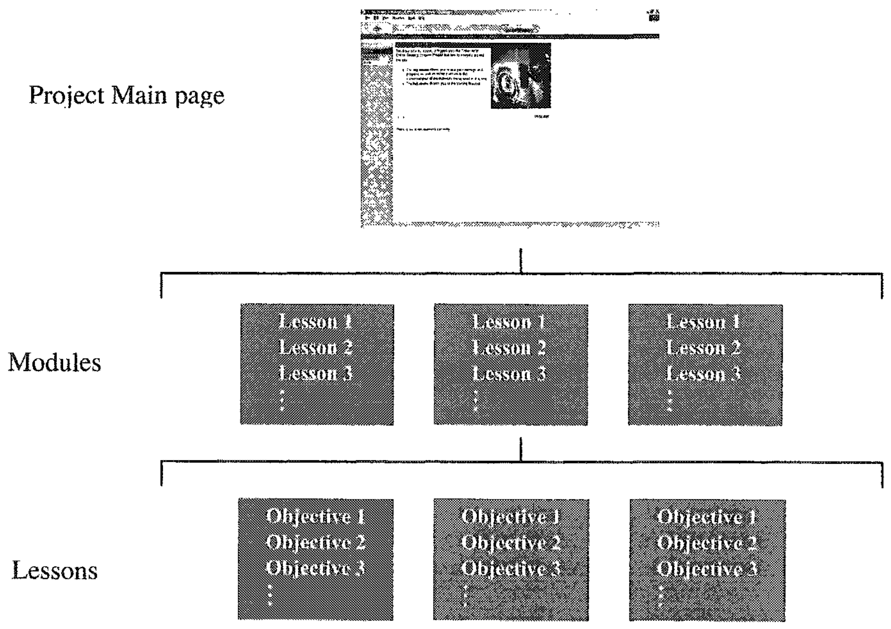


Figure 1.3: Project structure

The C-135 Aircraft Maintenance Instruction project gives detailed instructions on the correct maintenance procedures for pressurization on the KC-135 series aircraft. This project has a similar learning system structure to that of Environmental Engineering with the main difference being that each module represents the main technical maintenance tasks required for the KC-135. A unique feature of this project is the incorporation of three-dimensional visualization into the learning environment. Three-dimensional animations and simulations are developed to depict the actual situation of the maintenance procedure.

Research Outline

This research covers the investigation of the latest technology in delivering a more effective interactive web-based environment for engineering education and technical training. The next chapter presents a literature review that provides a general introduction to the relevant areas of web-based instruction, where in several learning systems developed through course management software and delivered as readymade packages are investigated to analyze the merits and demerits to developing an improved web-based training system. In chapter 3, the fundamental elements of the interactive web-based environment are identified. These elements are derived from the expectations of the end users.

Chapters 4 and 5 are the key chapters of this research. Chapter 4 discusses the development of interactive web-based training system for Tinker AFB, namely the EM/C135 System since this is the base system for the demonstration environments. In continuity with chapter 4, chapter 5 presents the development, implementation, and functionality of interactive web-based learning environments that are developed for an illustration purpose, including the CSD System and the 3D Dynamics System.

Chapter 6 elucidates the various technologies used in web-based training systems, and discusses the benefits of using commercial software in the development of web-based training systems. Additionally, the technology aspects of the actual learning environment are described. This includes an explanation of the integration techniques used to incorporate the media components in the environment. Finally, the concluding chapter provides the summary, conclusion, accomplishments, and contributions of the research. In this chapter, an assessment of the effectiveness of the training system is presented in order to support the usefulness of this research. Additionally, the final chapter discusses recommendations for future improvement and research in improving web-based training systems.

CHAPTER 2: LITRATURE REVIEW

Research has been conducted both in industry and educational institutions to develop environments for web-based education and training. As organizations are becoming more cost and time conscious, they are trying to find out the most efficient ways to educate and train their participants to increase the awareness of their responsibilities. Since numerous papers on web-based education and training have been published during the past several years, a literature review is done to explain how the Internet can be used as a tool in education and training.

In the fall of 1996, there was a study sponsored by the California Department of Education [11] results of this study concluded that the Internet and the web would play multiple important roles in higher education. They were expected to improve learning and teaching, sophisticate the creation of instruction and learning materials, develop educational communities, compete with new educational providers, and support policy making and planning. Envisioning these potential advantages, both universities and industries became leading forces of investment in research and development of the Internet infrastructure technologies for educational purposes. With the advent of the web, universities started to employ the Internet as a new educational tool to assist students by providing this service. There are several ways to use the web for education or training; an online learning system is one of the most admired applications.

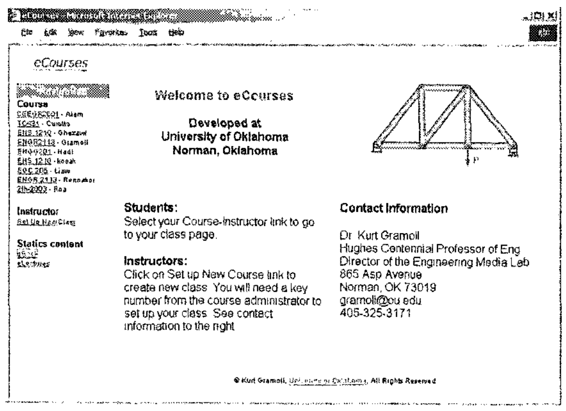
Several organizations are building online education and training systems to target specific groups of users. Each system distinguishes itself from the other with unique capabilities and features. Uniqueness varies from developmental, instructional, and administrative features available to student tools in the learning environment. This literature survey compares online learning systems developed using WebCT, Blackboard, and customized WBT (Web-Based Training) systems to target specific population groups.

Many educational institutions have begun to use electronic media to deliver the course material online. Hillman and Wells [12] mentioned the implementation of online courses with WebCT at Mississippi State University as an experimental project and since then, they have started moving towards offering electronic courses. The response and positive feedback from the students and instructors for the online courses developed surprised the administration with success. Currently, the system is widely used by professors at many universities as a supplementary teaching tool. WebCT, a commercial tool, is widely used to provide a systematic way to develop online courses [13]. Strenth [14] developed a Civil Construction class on the Internet using Blackboard. This course was made possible for students to participate during the spring of 2000 at Pittsburg State University. The responses from the students were varied; those students who finished the assignments and coursework before the scheduled time were satisfied, but some of the students who tried to complete the assignments at the last minute were frustrated as a result of the submission problems. The majority of the students, however, had inquired about another summer Internet class. Referring to a Blackboard press release, current developments in Blackboard as an instructional management system have reached more than 300,000 users at around 3,300 institutions in 70 countries [15].

The limitation of using Blackboard and WebCT is the inability to create a customized framework and structure to support specific course requirements, capabilities, and targets. Blackboard and WebCT provide only a general framework structure for development of web-based training systems. Though the flexibility to build the web-based courses from scratch in a short period of time is a unique feature; courses developed using Blackboard lacked the ability to incorporate an interactive teaching environment. Support tools for technical education, such as visualization, real-time evaluation, user tracking ability, and collaborative learning environments, are not supported in either system. Blackboard and WebCT are not engineering and technical centric. This lack of ability affects engineering education and technical training because interactivity and collaboration help students better understand course materials presented on the Internet by allowing students to explore engineering concepts [16, 17, 18]. This concept of collaborative learning was supported with various customized WBT systems such as Virtual FlyLab, Virtual Earthquake [19], and Virtual City [20, 21]. These systems incorporated many positive features available in Blackboard and WebCT and also reduced their limitations in an online learning environment.

At the University of Tennessee, a graduate level Finite Element course was offered as a live, video-streamed lecture over the web [22, 23]. This provided an opportunity for distance and off-campus local students with the Internet connection to take classes with on-campus students. To minimize the complexity of course delivery in distance education, the Georgia Institute of Technology recently utilized the Internet as a new medium to present content to students at remote locations [24]. This instant online distribution of course content increased the speed of delivery and supported the flexibility to update, modify and change the distributed content.

In Engineering Media Lab (EML) at the University of Oklahoma, several online learning systems have been developed to assist students in engineering education. An engineering project called eCourses was developed to provide a foundation in basic Statics and Dynamics classes through the use of laptop computers and the web [25, 26]. The web was used for the distribution of all course content, including an eBook, homework, examples, quizzes, solutions, and lectures. The lectures were recorded by a video camera, compressed into Flash streaming video format, and placed on the server. This was done to provide the flexibility to students to view missed lectures or review lectures again through the web at their own convenience. The system also included a basic collaborative drawing tool to assist students during online office hours. Figures 2.1a and 2.1b illustrate sample screens of eCourses.



Welcome to eCourses

Developed at
University of Oklahoma
Norman, Oklahoma

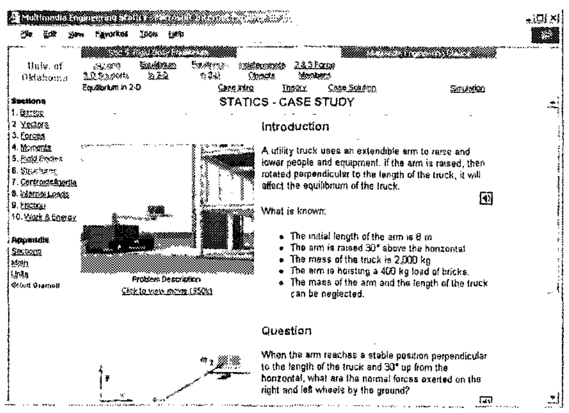
Students:
Select your Course-Instructor link to go to your class page.

Instructors:
Click on Set up New Course link to create new class You will need a key number from the course administrator to set up your class See contact information to the right

Contact Information

Dr Kurt Gramoll
Hughes Centennial Professor of Eng
Director of the Engineering Media Lab
865 Asp Avenue
Norman, OK 73019
gramoll@ou edu
405-325-3171

© Kurt Gramoll, University of Oklahoma, All Rights Reserved



STATICS - CASE STUDY

Introduction

A utility truck uses an extendible arm to raise and lower people and equipment. If the arm is raised, then rotated perpendicular to the length of the truck, it will affect the equilibrium of the truck.

What is known:

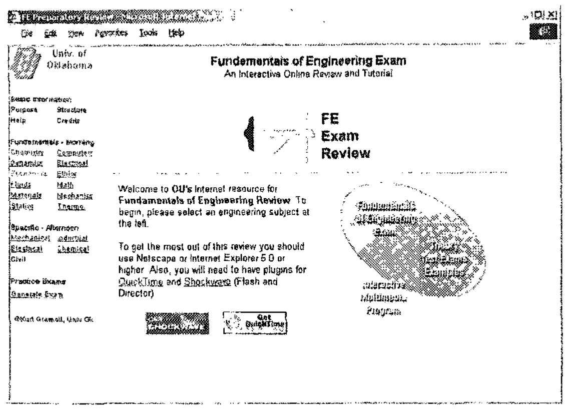
- The initial length of the arm is 8 m
- The arm is raised 30° above the horizontal
- The mass of the truck is 2,000 kg
- The arm is hoisting a 400 kg load of bricks.
- The mass of the arm and the length of the truck can be neglected.

Question

When the arm reaches a stable position perpendicular to the length of the truck and 30° up from the horizontal, what are the normal forces exerted on the right and left wheels by the ground?

Figures 2.1a and 2.1b: Sample screens of eCourses

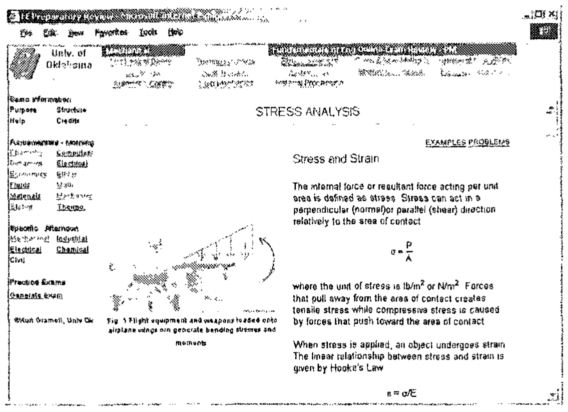
The Fundamentals of Engineering Exam Review project was a paradigm that uses the web as a delivery medium to offer online review [27]. The purpose of the project was to provide flexible, easily accessible, cost-effective and interactive review materials with integration of multimedia technology such as graphics, animations, audio, video and simulations for students and engineers to prepare for the Fundamentals of Engineering Exam (FE) during their quest to obtain their Professional Engineering License. In this online review, not only the basic review topics are covered but also the online exam is presented with randomly generated problems, to simulate the real time exam. Figures 2.2a and 2.2b illustrate sample screens of Fundamentals of Engineering Exam Review.



Fundementals of Engineering Exam
An Interactive Online Review and Tutorial

Welcome to OU's Internet resource for **Fundamentals of Engineering Review**. To begin, please select an engineering subject at the left.

To get the most out of this review you should use Netscape or Internet Explorer 5.0 or higher. Also, you will need to have plugins for QuickTime and Shockwave (Flash and Director)



STRESS ANALYSIS

Fig. 1 Flight equipment and weapons loaded onto airplane wings can generate bending stresses and moments

Stress and Strain

The internal force or resultant force acting per unit area is defined as stress. Stress can act in a perpendicular (normal)or parallel (shear) direction relatively to the area of contact

$$\sigma = \frac{P}{A}$$

where the unit of stress is lb/in^2 or N/m^2. Forces that pull away from the area of contact creates tensile stress while compressive stress is caused by forces that push toward the area of contact

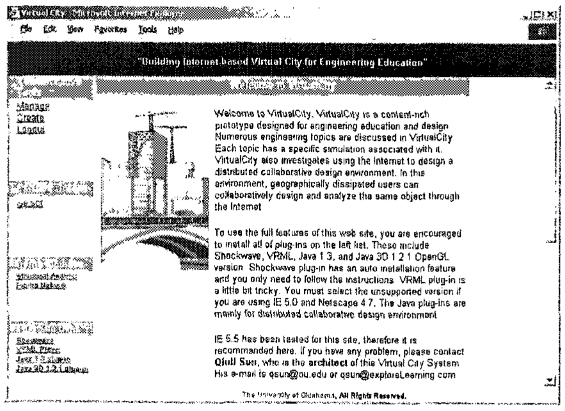
When stress is applied, an object undergoes strain. The linear relationship between stress and strain is given by Hooke's Law

$$\varepsilon = \sigma/E$$

Figures 2.2a and 2.2b: Sample screens of Fundamentals of Engineering Exam Review

Although it is generally accepted that the use of simulations in engineering education is beneficial to students, simulations are not widely used because the advantages gained by the user group population using the media does not outweigh the cost of development and distribution. With the incorporation of the Internet technologies, these problems can be minimized. Two interactive web-based applications called Virtual FlyLab and Virtual Earthquake were developed for learning science [19]. An innovative use of Internet technology for education employed Virtual Reality Modeling Language (VRML) to teach design over the Internet. The design module interfaced VRML with the Practical Extraction and Reporting Language (PERL). Another creative application used Java and VRML to teach machine kinematics [28]. Traditionally, machine kinematics was taught using drafting tools. With the introduction of computer graphics and the Internet, visual demonstration of concepts in machine kinematics by using three-dimensional models became a more efficient process than the traditional method. ExploreLearning [29] (www.explorelearning.com) is an interactive content creator that has developed exemplary multimedia simulations to help teach K-14 students a full range of subjects, from Algebra to English. The main objective of simulation developments is to revolutionize the way teachers teach and students learn with the use of simulations that provide hands-on learning combined with assessment tracking to result in student and teacher success. Simulations are mainly developed using Director and other technologies.

Another project that allows the interactivity and collaboration is the Virtual City project, developed in EML at the University of Oklahoma. The objective of this project was to provide guidance to students in civil engineering to better understand and integrate the theoretical concepts involved with the process of design and analysis through an interactive and distributed collaborative process. This technology employed VRML, Java, and Java 3D technologies to implement the major component of the system, which is three-dimensional visualization [20, 21]. Figures 2.3a and 2.3b illustrate sample screens of Virtual City.

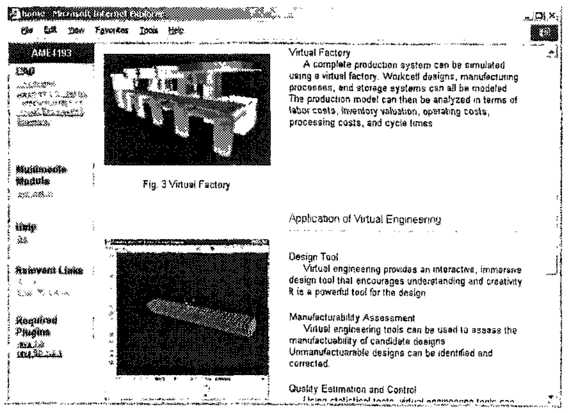

File Edit View Favorites Tools Help
"Building Internet-based Virtual City for Engineering Education"
Manage
Create
Logout

Welcome to VirtualCity. VirtualCity is a content-rich prototype designed for engineering education and design Numerous engineering topics are discussed in VirtualCity Each topic has a specific simulation associated with it. VirtualCity also investigates using the Internet to design a distributed collaborative design environment. In this environment, geographically dissipated users can collaboratively design and analyze the same object through the Internet

To use the full features of this web site, you are encouraged to install all of plug-ins on the left list. These include Shockwave, VRML, Java 1 3, and Java 3D 1 2 1 OpenGL version Shockwave plug-in has an auto installation feature and you only need to follow the instructions VRML plug-in is a little bit tricky. You must select the unsupported version if you are using IE 5.0 and Netscape 4 7. The Java plug-ins are mainly for distributed collaborative design environment

IE 5.5 has been tested for this site, therefore it is recommended here. If you have any problem, please contact **Qiuli Sun**, who is the **architect** of this Virtual City System His e-mail is qsun@ou.edu or qsun@exploreLearning com

The University of Oklahoma, **All Rights Reserved.**



Virtual Factory
A complete production system can be simulated using a virtual factory. Workcell designs, manufacturing processes, and storage systems can all be modeled The production model can then be analyzed in terms of labor costs, inventory valuation, operating costs, processing costs, and cycle times

Fig. 3 Virtual Factory

Application of Virtual Engineering

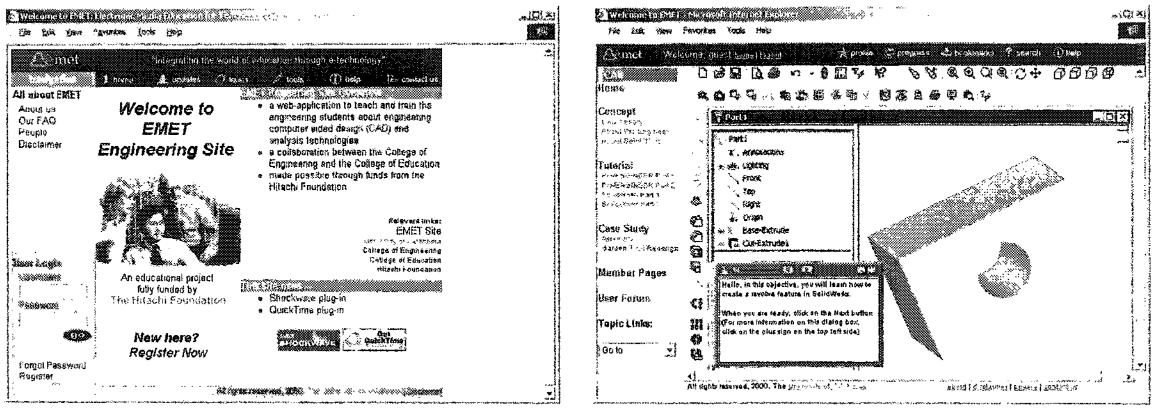
Design Tool
Virtual engineering provides an interactive, immersive design tool that encourages understanding and creativity It is a powerful tool for the design

Manufacturability Assessment
Virtual engineering tools can be used to assess the manufactuability of candidate designs Unmanufactuarable designs can be identified and corrected.

Quality Estimation and Control

Figures 2.3a and 2.3b: Sample screens of Virtual City

The EMET-Engineering project is an interactive web-based learning environment that heavily employs simulation technique as a primary element. This environment, which developed in EML as well, was designed to help teaching and training engineering students about Computer-aided Design (CAD) and analysis technologies. Simulation technique is implemented using Shockwave technology, which is developed through Macromedia Director to imitate capabilities of other engineering applications such as Pro/ENGINEER and SolidWorks. Because of the mimicking capabilities of Director, users do not need to buy or install any program on to their local computers to learn the software [30]. Figures 2.4a and 2.4b illustrate sample screens of EMET-Engineering.


Welcome to EMET Engineering Site
An educational project fully funded by The Hitachi Foundation
New here? Register Now
Concept
Tutorial
Case Study
Member Pages
User Forum
Topic Links:

Figures 2.4a and 2.4b: Sample screens of EMET-Engineering

Wallace and Mutton [31] compared the effectiveness of web-based lecture versus classroom-based lecture. The results showed that students using the web-based lectures performed better than students taking classroom-based lectures. Although the results and conclusions of the experiment were based only on one topic, it rejected the hypothesis that the two groups perform equally. In another similar experiment conducted also by Wallace [32], the performance of two groups of students was investigated. The first group was made to prepare for class using web-based materials, while the other was given a traditional classroom-based education with the constraint of using information provided only during the class. In the first group, students were prepared for class with both web-based materials and the lecturer. After preparation, the students of these two groups were made to complete an assignment. It was inferred that the average grade performance of the first group was better than that of the second group. The experimental finding suggested that traditional classroom time could be used for higher value-added activities such as mentoring and experiential activities during a course by integrating and organizing classroom-based teaching with the web-based delivery of material. At the University of Missouri-Rolla, the effectiveness of an online graduate engineering management course was investigated [33]. The results showed that more than half the students believed that taking an online course was a significant and effective learning process. Although little documentation of the effectiveness of web-based courses is available, more research is needed to identify why web-based courses help learning.

Currently, there are several educational learning systems that incorporate a variety of electronic media over the Internet. Three-dimensional visualization has played a major role in these existing technical learning environments. However the method of implementing three-dimensional visualization can still be improved to deliver the content more conveniently. Thus one of the major goals of this research is to identify and illustrate the implementation of new techniques to provide an improved environment.

CHAPTER 3: ESSENTIAL ELEMENTS OF THE ENVIRONMENT

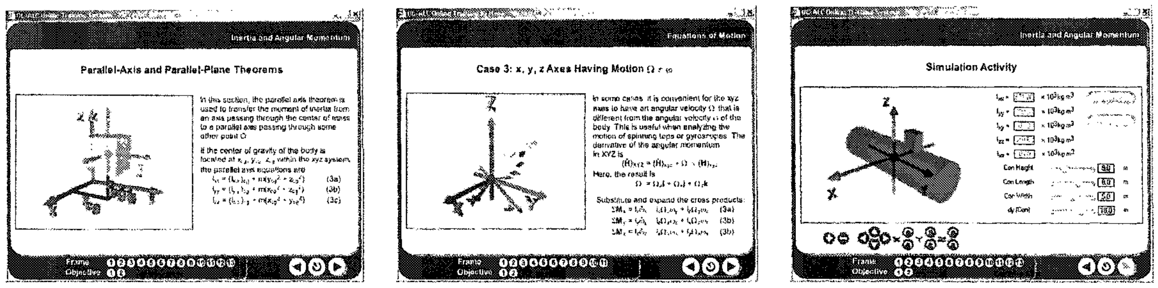
To develop an interactive web-based environment for engineering education and technical training, it is important to identify its fundamental features derived from the expectations of the end users. Different researchers may have different viewpoints, but this research focuses on the discussion of essential features such as interactivity, three-dimensional visualization, evaluation, user tracking ability, and server-side database and web-based administrative system. These fundamental elements should be ideally included in interactive web-based environments for engineering education and technical training that are expected from the viewpoint of the end users. This is because with these key elements, interactive web-based environments for engineering education and technical training are more effective. This chapter discusses how these elements associate with interactive web-based environments for engineering education and technical training.

Interactivity

Interactivity in a learning environment helps students better understand course materials presented on the Internet by allowing students to explore engineering concepts [8]. A web-based learning environment should not only present the course materials in text, graphics, sound, and video, but should also create an interactive environment in which students can explore and obtain immediate feedback. Interactivity needs to be maintained throughout the learning process to make the system participation intensive so that the concentration of the student is focused on the subject. Enabling the student to learn through exploration and investigation along with the content is also important so that comprehending through experimentation is supported.

Three-dimensional Visualization

To be successful in demonstrating engineering and technical concepts, the best way is to present users with three-dimensional models [21, 34, 35]. This can eliminate many words describing the complicated concepts and give the user a direct impression of the real model. Incorporating three-dimensional visualizations in interactive simulations can be used to assist the user in his or her learning progress by using virtual prototype models that would be available in reality. Figures 3.1a, 3.1b, and 3.1c illustrate sample three-dimensional visualizations such as graphics, animation, and simulation.


Inertia and Angular Momentum
Parallel-Axis and Parallel-Plane Theorems
Equations of Motion
Case 3: x, y, z Axes Having Motion Ω ≠ ω
Inertia and Angular Momentum
Simulation Activity
Frame
Objective

Figures 3.1a, 3.1b, and 3.1c: Graphics, animation, and simulation

Evaluation

In order to evaluate the student's understanding of the material, a set of questions and a variety of activities that require action by the students are designed. This provides feedback to the student and administrator about the comprehension progress by the end of the learning block. The questions cover the basic concepts that the student should have acquired throughout that particular learning block. Besides the typical questions, there are various simulation activities throughout the learning system. These activities require the student to participate by doing something in order to measure his or her progress.

User Tracking Ability

A web-based learning environment should not only present the course materials, but also have the ability to track student information. Information tracking is necessary for two purposes: first, it is required to assist students to learn better by making their own assessments; secondly, it is needed for administrators and supervisors for reviewing and evaluating students. An additional advantage of tracking such information is that it provides data regarding the usability of the system over a period of time. The capability to keep track of the user is not limited to only two-dimensional simulation activities, but also to three-dimensional.

Server-based Database and Web-based Administrative System

Server-based database is considered significant because of the necessity to support administrators in managing a system effectively. In the learning system, all relevant student information is tracked and recorded in a database that resides on the server. The delivery of learning materials and the tracking of student progress are also controlled using the same database. All information is stored within the database and is critical to the delivery of the learning materials. Furthermore, a web-based administrative system is needed in order to accommodate the administrator to manage the system in a simple manner anywhere Internet is accessible. This will allow the administrator to monitor students and supervise the overall system conveniently through a web-based interface.

Implemented Elements in the Demonstration Environments

The two demonstration environments developed are the 3D Dynamics System and the CSD System (CSD stands for the *Constant Speed Drive*). The 3D Dynamics System is considered to be an expanded section from the EM/C135 System. The two differences are that the former material content concentrates on the topic of 3D Dynamics and extensively uses three-dimensional visualization to achieve the objectives of this research. However, different researches may have different viewpoints and the CSD System, which is the other demonstration environment, focuses only on the integration of interactivity, three-dimensional visualization, and evaluation. These elements are the minimum requirements to accomplish the main objectives of this research and fulfill the essential features that are needed to be presented in the demonstration environment, as defined by Tinker Air Force Base (AFB).

CHAPTER 4: THE DEVELOPMENT OF EM AND C135 SYSTEM

The concept of web-based training is being widely adopted both in profit and non-profit organizations. Tinker Air Force Base (AFB), a non-profit governmental installation, has also adopted this method of training in order to increase the proficiency of its personnel in complying with environmental engineering regulations and to improve personnel's efficiency at performing various aircraft maintenance operations. The high security and confidentiality of information has forced Tinker to locate training within its base; therefore, in order to resolve the high security issue, the Internet is used as a tool to deliver training. By using the Internet as delivery medium, the impact of secondary components, such as convenient scheduling, active participation, attendance, participation, and time, which affect the organization of training sessions, are also minimized. This chapter describes the incorporation of three-dimensional components with other media elements in the interactive web-based training projects for both military and civilian personnel at Tinker AFB. These projects include training material for both Environmental Engineering and C-135 Aircraft Maintenance Instruction.

The purpose of the Environmental Engineering project is to train and guide the Air Force Base personnel to comply with the standard requirements of various NESHAP (National Emission Standard for Hazardous Air Pollutants) and environmental regulations. Understanding and complying with NESHAP is fundamental for any person who performs duties on any air base in the US. The C-135 Aircraft Maintenance Instruction project gives detailed instructions on the correct maintenance procedures for pressurization on the KC-135 series aircraft.

Engineering knowledge and technical skill are necessary for project compliance and routine activities. The specific need to train personnel to comply with regulations placed this project on center stage at Tinker AFB. Training can also be used to improve efficiency and to attain specific technical skills. This training is focused towards guiding both engineers and technicians to become more proficient and to attain the objectives of their projects by complying with the technical aspects.

The learning system developed and implemented for Tinker AFB, uses various web-based technologies to provide a platform for training. One advantage of using the Internet is the capability to track and monitor users from remote locations within and outside the base. Statistical information on users' progress, such as actual time spent on each objective and the number of times visited on any assigned topic, are monitored and recorded in an online database. The information tracked is used for two purposes. First, it is used to assist students learn better by allowing them to make their own assessments. Second, it is made available to administrators and supervisors for review and evaluation. An additional advantage of tracking such information is that it provides data regarding the usability of the system over a period of time.

In order to intensely understand the details of designing this learning system, a brief overview of the various topics involved in NESHAP and C-135 is given. Further, the structure of the learning system is explained in the next section. Also, the learning system organization, functional operation of the lessons, information tracking, and the built-in administrative capabilities are discussed later.

Topics Covered in the Learning System

The learning system includes a variety of training material for both environmental engineering and aircraft maintenance instruction. These two topic areas are divided into subtopics that refer to each main topic covered in order to increase the learning progress of users. There are six NESHAP requirements and environmental regulations involved in Environmental Engineering training. To meet these requirements and regulations, this project is categorized into modules, where each module is devoted to a specific regulation applied at Tinker AFB. The Tinker AFB Aerospace NESHAP module provides an overview of Aerospace NESHAP, along with maintenance, responsibility for compliance, cleaning operations, and painting procedures. The Tinker AFB Introduction to the Clean Air Act module provides an introduction to the act including results of air pollution, direct outcomes of air pollution, and how can we limit air pollution. The Tinker AFB Halogenated Solvent NESHAP provides an introduction to Halogenated Solvent NESHAP, along with operations, maintenance requirements, and responsibility for compliance. The Tinker AFB Chrome Electroplating NESHAP module provides an introduction to Chrome Electroplating NESHAP including maintenance, operations requirements, and responsibility for compliance. The Pollution Prevention module provides an introduction to the pollution prevention process and focuses on compliance through pollution prevention. The Solid Waste Management module provides an introduction to and an overview of Solid Waste Management, including discussions about existing program and potential improvements, applicable rules, and regulations.

There are four maintenance procedures for the pressurization of the C-135 series aircraft inside the C-135 Aircraft Maintenance Instruction project. The project is categorized into modules to support these various procedures and each module is devoted to an individual procedure required for maintaining pressurization. The Cabin Pressure Control System Familiarization module provides information about component description, system operations, and the test equipment required. The Fuselage Pressure Testing module covers three required steps for proof test, including preparations, procedures, and post procedures. The Cabin Pressurization module describes operational checkout preparations, precautions, and procedures, while the Cabin Pressure Control System Malfunction Analysis module discusses analysis procedures for cabin pressure control system malfunctions.

Learning System Structure

In order to manage and organize materials efficiently, a modular and hierarchical structure is incorporated into designing the learning system. The structure is designed in close relationship to that of an academic curriculum wherein each level can be compared relatively to a traditional education system. Inside each project (Environmental Engineering and C-135 Aircraft Maintenance Instruction), four levels are maintained to hierarchically structure the contents. The breakdown structure is organized into modules, lessons, objectives, and frames as illustrated in Figure 4.1. Each project is categorized into modules, where each module is dedicated to a single topic. Each module consists of several individual lessons, and then each lesson is subdivided into objectives, which are the basic learning blocks.

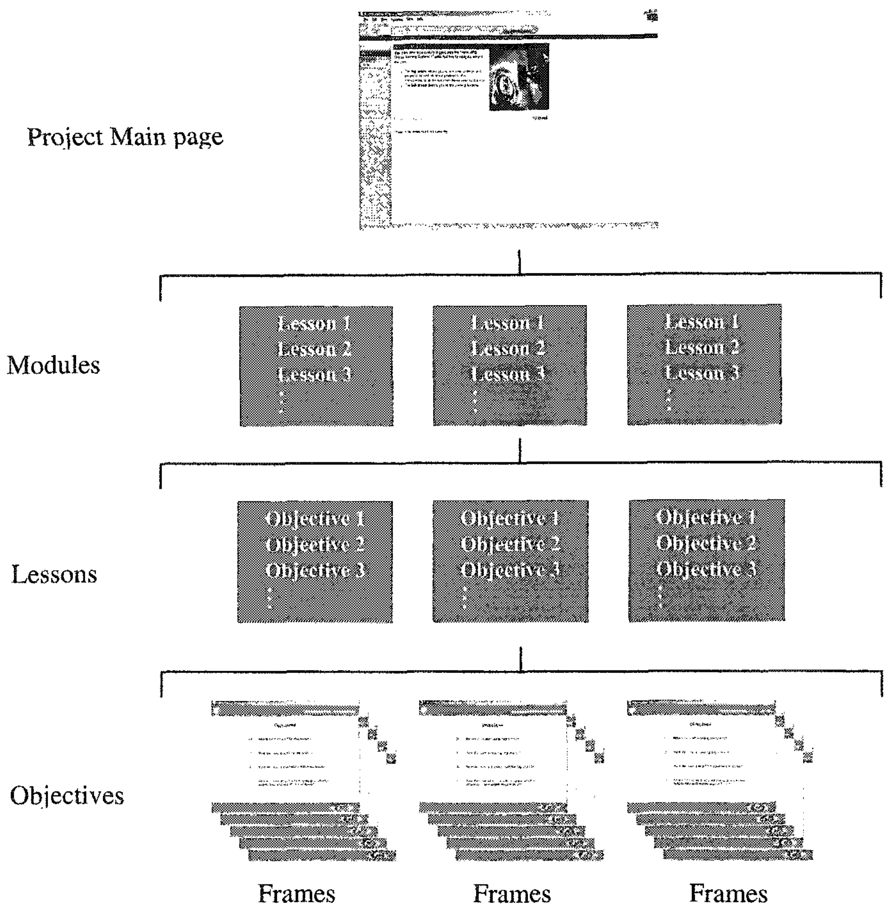


Figure 4.1: Organization of modules, lessons, and objectives

The contents in each objective are displayed in separate frames. Each frame includes at least two media components. Though each frame displays specific content, they are self-contained in a single environment for each objective; this allows users to learn the material in pieces and then move on to the next concept. Furthermore, if a user stops at a given frame, it is best to have the user start back at the beginning of the objective so that the concept in that objective can be reviewed from the beginning to the end.

Types of Users

Information organization is an important aspect for delivery, since the front-end interface displays information based on the type of user accessing the system at that time. Users are categorized into four types in this system, including administrative users, manager users, regular users, and guest users. Administrators and managers are also classified as supervisors and regular and guest users are categorized as users. Although they are different types of users accessing through the same interface, the privileges and features accessible for each user are identified after the checking procedure by the database. A three-level management system as shown in Figure 4.2 is used to design the hierarchical structure of control in the system.

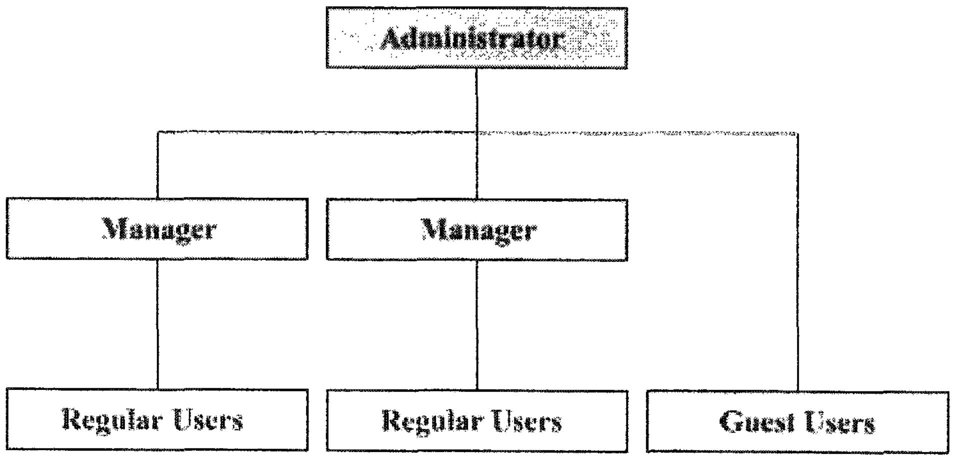


Figure 4.2: Management level

Administrators can manage all the administrators, managers, regular users and guests; managers can administer all managers and regular users in their group; regular users and guests have no control over others. In other words, administrative users are power users who have the highest privileges to access management features and options for administration purposes in the system. Since there are several administrative features incorporated, a detailed discussion is presented later in this chapter. In this system, multiple administrators can be set to ease the general management and website maintenance.

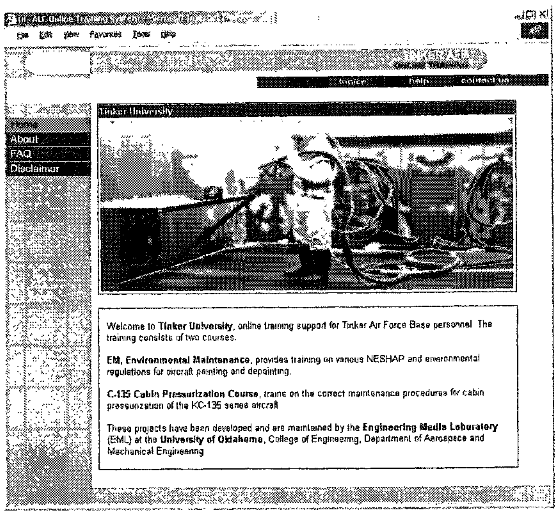
The second level in the three-level system is manager users. *Managers* have less administrative control when compared to administrators. Managers are provided with capabilities similar to administrative users. However, their privileges are limited only up to supervising capabilities, such as setting up and managing users only in their group. Therefore, managers can use the same administrative features available for administrators to control regular users in their department and all guests, but no changes related to administrators can be made by managers. In other words, managers have restricted power over the training system. Several managers can be set up to supervise personnel from their own workforce. The purpose of having managers administrate the system and having users beside administrators is to decrease administrators' workload since there are numerous users involved in this training system. These users can be separated into a variety of groups that can be easily managed by managers who are only responsible for a particular group.

Finally, the lowest level of users includes regular users and guest users. These users are given no administrative privileges but can access the training content and support features. *Regular users* are the majority type of users in the training system. Administrators or managers assign a single lesson group or multiple lesson groups to them. This limits normal users' access to only lessons assigned to that group when they access the system and they have to study those lessons and objectives in sequence in order to complete the learning process.

Guest users are classified as anonymous users visiting the website. They are users not set up as students either by an administrator or a manager. For these types of users, though entry is not restricted, access to content is limited for security issues in the organization. They do not belong to any of the lesson groups and they have to view all the modules sequentially. However, as one objective is completed, data about the score, question number, module number, lesson number, objective number, user name, cumulated time (in minutes) spent in the objective and last accessed date will be updated to the database in the same way as regular users and supervisors if they take any lessons. As guests, their accounts will expire after a given duration that can be set by administrators or managers using the *Preferences* feature, which is one of the administrative features discussed later in this chapter. When an account is expired, all information related to the particular guest will be deleted from the system. Additionally, administrators or managers can change a guest's account into another type of account like administrator, manager or normal user as they desire.

Learning System Organization

In order to evaluate the effectiveness of this training system via the Internet, tinker.ou.edu hosted at the University of Oklahoma is dedicated the delivery website. Each project can be accessed by selecting one of the topic areas provided in the *system main page* as illustrated in Figure 4.3. Currently, there are two topic areas; the first is Environmental Engineering, and the second is C-135 Aircraft Maintenance Instruction.



Welcome to **Tinker University**, online training support for Tinker Air Force Base personnel The training consists of two courses.

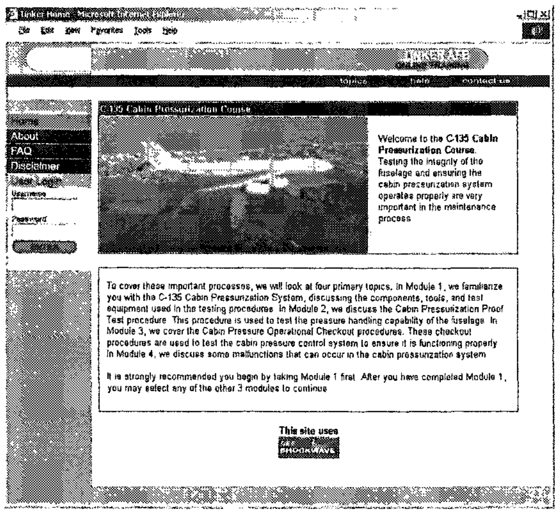
EM, Environmental Maintenance, provides training on various NESHAP and environmental regulations for aircraft painting and depainting.

C-135 Cabin Pressurization Course, trains on the correct maintenance procedures for cabin pressurization of the KC-135 series aircraft

These projects have been developed and are maintained by the **Engineering Media Laboratory** (EML) at the **University of Oklahoma**, College of Engineering, Department of Aerospace and Mechanical Engineering

Figure 4.3: System main page

After users select the project, the *project introduction page* as shown in Figure 4.4 is reached. This page provides a brief preface about each project available in the system. The system is made flexible in order to support the expandability needed to add additional topics without difficulty. In order to restrict distribution material and to protect confidentiality of information, a secure system with a login is implemented; however, Tinker can activate a "guest" account if they wish to allow others access to the material.


Tinker Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
topics
help
contact us
C-135 Cabin Pressurization Course
Home
About
FAQ
Disclaimer
User Login
Username
Password

Welcome to the **C135 Cabin Pressurization Course**. Testing the integrity of the fuselage and ensuring the cabin pressurization system operates properly are very important in the maintenance process

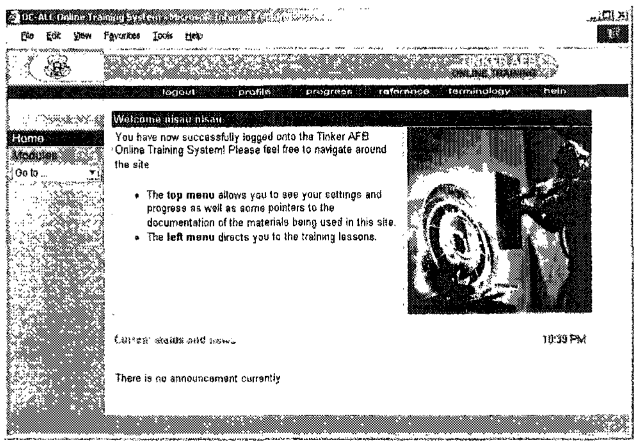
To cover these important processes, we will look at four primary topics. In Module 1, we familiarize you with the C-135 Cabin Pressurization System, discussing the components, tools, and test equipment used in the testing procedures In Module 2, we discuss the Cabin Pressurization Proof Test procedure This procedure is used to test the pressure handling capability of the fuselage In Module 3, we cover the Cabin Pressure Operational Checkout procedures. These checkout procedures are used to test the cabin pressure control system to ensure it is functioning properly In Module 4, we discuss some malfunctions that can occur in the cabin pressurization system

It is strongly recommended you begin by taking Module 1 first After you have completed Module 1, you may select any of the other 3 modules to continue

This site uses

Figure 4.4: Project Introduction page

The login procedure is the primary permission-checking tool used to access and enter the system. This mechanism is located in each *project introduction page.* After users successfully enter the project, the *project main page* as shown in Figure 4.5 appears.



Welcome nisau nisau

You have now successfully logged onto the Tinker AFB Online Training System! Please feel free to navigate around the site

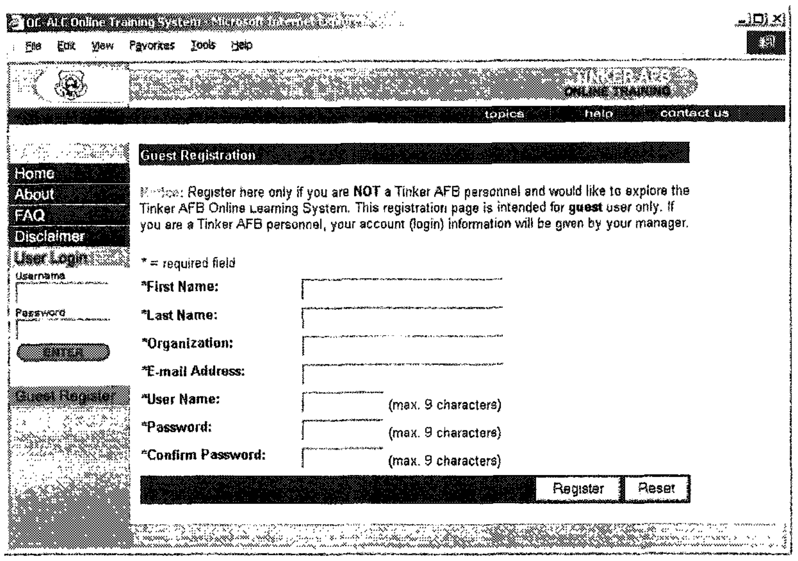
- The **top menu** allows you to see your settings and progress as well as some pointers to the documentation of the materials being used in this site.
- The **left menu** directs you to the training lessons.

Current status and news 10:39 PM

There is no announcement currently

Figure 4.5: Project main page

An active user has to enter the username and password that was set up by an administrator or manager account. This identification allows the system to track and record the progress of each user. The guest account is not included in any report for Tinker personnel. The system administrator has the ability to turn off the guest account option and lock out the system from the general public. Guest users can access the system using their username and password received after completion of the *Guest Registration form* as shown in Figure 4.6.


OC-ALC Online Training System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
TINKER AFB ONLINE TRAINING
topics
help
contact us
Home
About
FAQ
Disclaimer
User Login
Username
Password
ENTER
Guest Register

Guest Registration

Notice: Register here only if you are **NOT** a Tinker AFB personnel and would like to explore the Tinker AFB Online Learning System. This registration page is intended for **guest** user only. If you are a Tinker AFB personnel, your account (login) information will be given by your manager.

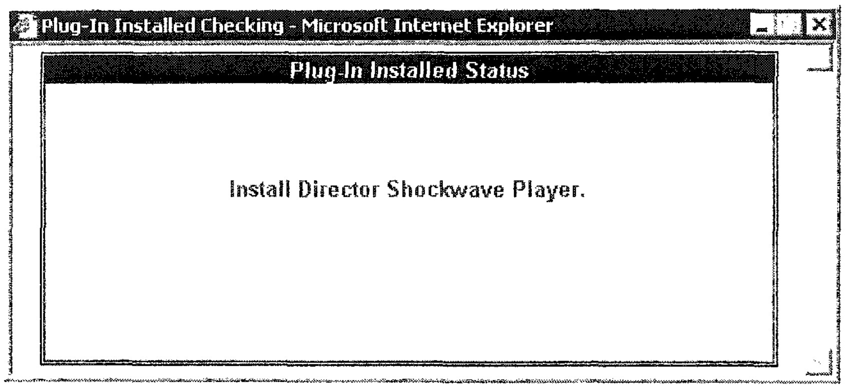
* = required field

***First Name:**		
***Last Name:**		
***Organization:**		
***E-mail Address:**		
***User Name:**		(max. 9 characters)
***Password:**		(max. 9 characters)
***Confirm Password:**		(max. 9 characters)

Register Reset

Figure 4.6: Guest registration form

Once a user enters the system successfully, the database performs checks to confirm the identity of the user, to determine their privileges, and to identify what lessons need to be completed. Not all lessons are open to all users. This feature allows Tinker to selectively target employees for specific training. Next, the installation status of the browser plug-ins is automatically checked and the user is given a status report in a popup window as shown in Figure 4.7 if any additional plug-ins are required. This popup window will not appear if all required plug-ins are already installed. Furthermore, the popup window provides links for users to download and install plug-ins. The system uses Shockwave (Director and Flash) media that require a browser plug-in before viewing the lessons.

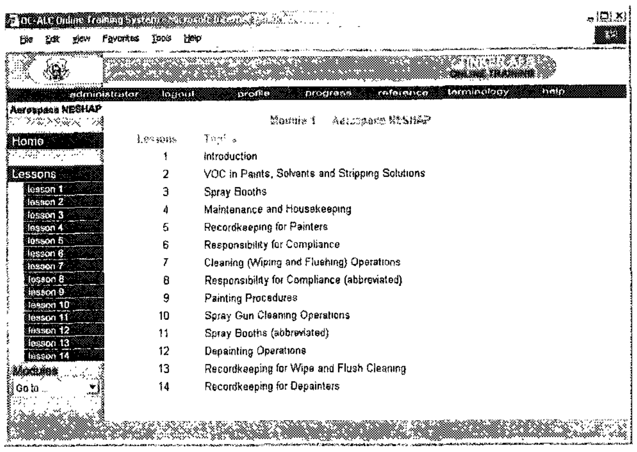


Install Director Shockwave Player.

Figure 4.7: Browser plug-in status feedback

In order to access a lesson, the navigation level inside the system is divided into various levels to match the organization of the modules (Figure 4.1). This process is designed to aid in information and content presentation, and guides users through the system conveniently and productively. The various stages in navigation are discussed briefly in the following paragraphs.

The first navigation level is the *personalized page* or *project main page* that appears after login. Two menus are provided at this level, as shown in Figure 4.5. The options in the top menu vary based on the type of user and are discussed in detail later. The menu on the left provides the option for users to navigate between modules by pulling down the dropdown menu. The individual lessons are inside the modules, as shown in Figure 4.8.

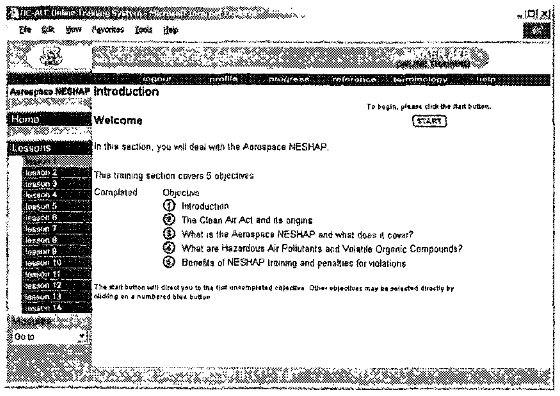


Module 1 Aerospace NESHAP

Lessons	Topics
1	Introduction
2	VOC in Paints, Solvents and Stripping Solutions
3	Spray Booths
4	Maintenance and Housekeeping
5	Recordkeeping for Painters
6	Responsibility for Compliance
7	Cleaning (Wiping and Flushing) Operations
8	Responsibility for Compliance (abbreviated)
9	Painting Procedures
10	Spray Gun Cleaning Operations
11	Spray Booths (abbreviated)
12	Depainting Operations
13	Recordkeeping for Wipe and Flush Cleaning
14	Recordkeeping for Depainters

Figure 4.8: Lesson Selection page

The organization of the lessons inside each module depends on the type of user and is controlled by the system administrator. Only those lessons assigned to a user are displayed and added to the left menu. The lesson numbers are also different between user groups due to variation in content. After a lesson is selected, a list of objectives in that lesson is displayed in the middle of the page, as shown in Figure 4.9. The left menu bar remains available if choosing another lesson is needed.



Introduction

To begin, please click the start button.

Welcome

In this section, you will deal with the Aerospace NESHAP.

This training section covers 5 objectives

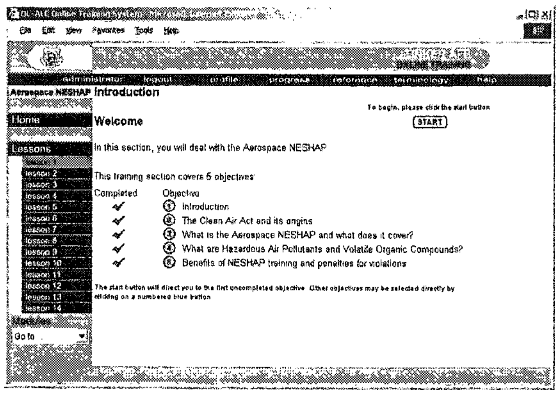
Completed Objective
① Introduction
② The Clean Air Act and its origins
③ What is the Aerospace NESHAP and what does it cover?
④ What are Hazardous Air Pollutants and Volatile Organic Compounds?
⑤ Benefits of NESHAP training and penalties for violations

The start button will direct you to the first uncompleted objective. Other objectives may be selected directly by clicking on a numbered blue button

Figure 4.9: Lesson Start page

The check mark next to each objective indicates if the objective has been viewed and completed. Any completed objective can be reviewed but new objectives must be taken in order. The Start button takes users to the beginning of the first objective not yet completed. If it is a user's first time in a given lesson, the Start button will begin at the first objective. This mechanism is implemented so that users are forced to go through the material in a linear process. All objectives must be done in order, but completed objectives can be reviewed in any order.

However, for administrators' accounts, the check marks are always activated no matter if this objective has been viewed or not (Figure 4.10), in order to provide administrators with the ability to access all objectives nonlinearly. This can help them to find the course objective quickly. Administrators can select any objective in any module and lesson anytime without going through other modules, lessons or objectives.



Aerospace NESHAP Introduction

To begin, please click the start button

Welcome

In this section, you will deal with the Aerospace NESHAP

This training section covers 5 objectives:

Completed	Objective
✓	① Introduction
✓	② The Clean Air Act and its origins
✓	③ What is the Aerospace NESHAP and what does it cover?
✓	④ What are Hazardous Air Pollutants and Volatile Organic Compounds?
✓	⑤ Benefits of NESHAP training and penalties for violations

The start button will direct you to the first uncompleted objective. Other objectives may be selected directly by clicking on a numbered blue button.

Figure 4.10: Objective description page for administrator

The actual delivery of lesson content is done in a new browser window that is opened when the Start button is clicked. The new window has all browser functions turned off so that users can only use the buttons provided inside the lesson frame to navigate. This allows the system to monitor where a user is at and what objective is being completed. The main browser window is still active for users to access help, profile, progress, reference, and terminology sections. An example of a new lesson window is shown in Figure 4.11.

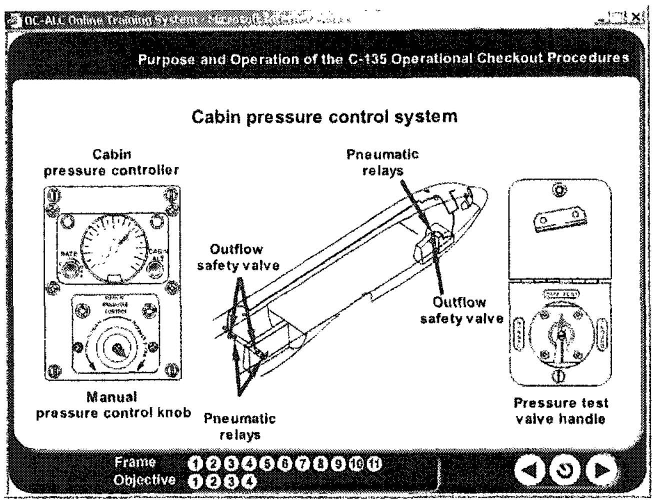
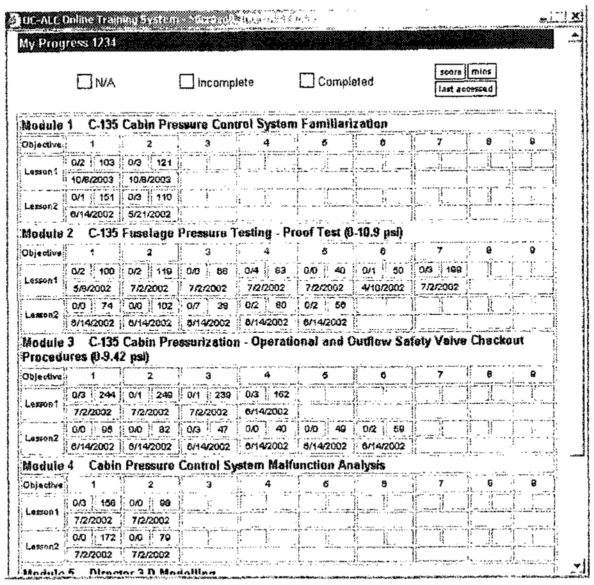

OC-ALC Online Training System
Purpose and Operation of the C-135 Operational Checkout Procedures
Cabin pressure control system
Cabin pressure controller
Pneumatic relays
Outflow safety valve
Outflow safety valve
Manual pressure control knob
Pneumatic relays
Pressure test valve handle
Frame
Objective

Figure 4.11: New Lesson browser window

Inside the main browser on the top menu option, support features to check progress, modify profile, acquire help, and view terminologies used are provided for all users. This feature is presented in a separate pop-up browser window, as shown in Figure 4.12. At this stage, users with administrative capabilities are provided with an additional link on the top menu of the interface in order to access the management features. JavaScript is used to control the size and location of the browser window. JavaScript is also used for targeting these multiple features accessed from the menu to the same separate pop-up browser window. This control is provided so that several windows are not opened, thus confusing users.



UC-ALC Online Training System

My Progress 1234

☐ N/A ☐ Incomplete ☐ Completed

score | mins
last accessed

Module 1 C-135 Cabin Pressure Control System Familiarization

Objective	1	2	3	4	5	6	7	8	9
Lesson1	0/2 103 10/8/2003	0/3 121 10/8/2003							
Lesson2	0/1 151 6/14/2002	0/3 110 5/21/2002							

Module 2 C-135 Fuselage Pressure Testing - Proof Test (0-10.9 psi)

Objective	1	2	3	4	5	6	7	8	9
Lesson1	0/2 100 5/8/2002	0/2 119 7/2/2002	0/0 86 7/2/2002	0/4 63 7/2/2002	0/0 40 7/2/2002	0/1 50 4/10/2002	0/3 199 7/2/2002		
Lesson2	0/0 74 6/14/2002	0/0 102 6/14/2002	0/7 39 6/14/2002	0/2 80 6/14/2002	0/2 56 6/14/2002				

Module 3 C-135 Cabin Pressurization - Operational and Outflow Safety Valve Checkout Procedures (0-9.42 psi)

Objective	1	2	3	4	5	6	7	8	9
Lesson1	0/3 244 7/2/2002	0/1 249 7/2/2002	0/1 239 7/2/2002	0/3 162 6/14/2002					
Lesson2	0/0 96 6/14/2002	0/0 82 6/14/2002	0/3 47 6/14/2002	0/0 40 6/14/2002	0/0 49 6/14/2002	0/2 59 6/14/2002			

Module 4 Cabin Pressure Control System Malfunction Analysis

Objective	1	2	3	4	5	6	7	8	9
Lesson1	0/3 156 7/2/2002	0/0 99 7/2/2002							
Lesson2	0/0 172 7/2/2002	0/0 79 7/2/2002							

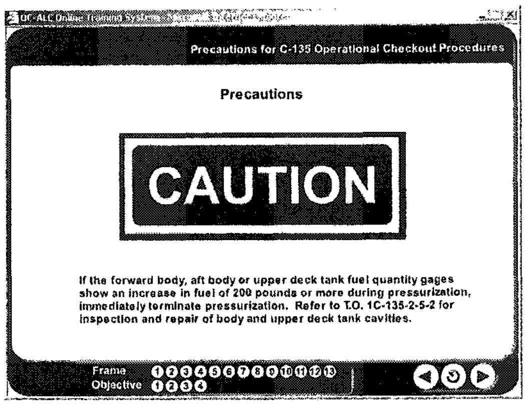
Module 5 Director 3 D Modelling

Figure 4.12: A separate browser shows user progress report

Lesson Operation

The primary focus of this learning system is to educate and to provide an opportunity for users to understand their responsibilities. To assist in this goal, it is important that the actual lesson navigation is simple. The level of control must also provide the option for users to review present content as well as any completed topic while in a given objective.

To focus users' attention on the material presented, a simple navigation system that can provide the required level of control is implemented. Three main functional navigation buttons are provided for users to control the operation of the lesson. They are the forward, repeat, and backward buttons. The navigation is located in the bottom right corner inside each lesson screen so that the buttons do not interfere dependent on the audio instructions that explain the concepts with the presentation material. So that users must view the contents inside each objective at least once before proceeding to the next concept, forward navigation is made in each frame. Users must wait until the instructions are completed before the forward button is activated. This attribute restricts users from quickly moving through the lesson without listening to and viewing the content. An example of a lesson frame, which also illustrates the basic user controls, is shown in Figure 4.13.


Precautions for C-135 Operational Checkout Procedures

Precautions

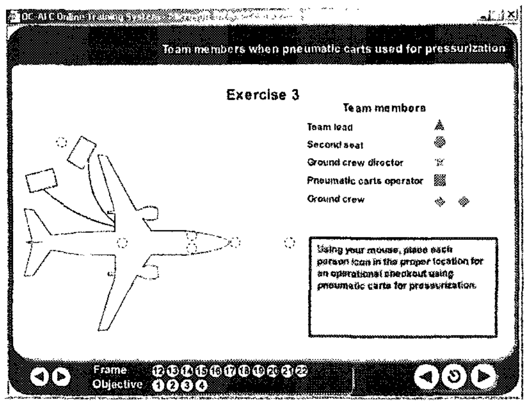
CAUTION

If the forward body, aft body or upper deck tank fuel quantity gages show an increase in fuel of 200 pounds or more during pressurization, immediately terminate pressurization. Refer to T.O. 1C-135-2-5-2 for inspection and repair of body and upper deck tank cavities.

Figure 4.13: Typical lesson frame or screen

In addition to the forward button, other functional navigation buttons include the backward and repeat buttons. These buttons are always activated. The backward button allows users to review the previous frames. The repeat button is located in between the forward and backward buttons because its functionality lies between the other two features. The repeat button is incorporated to provide users with the option to review the material inside a particular frame again before proceeding.

Moreover, frame buttons allow users the convenience of navigating to any specific previous frame. The numbers on the buttons show users exactly how many frames are included in a particular objective and how many objectives in this lesson. These buttons are incorporated with three different colors to indicate where users are at and the status of completion of users as well. The yellow buttons indicate the current position to users. The blue buttons indicate the frames and objectives that have been learned. The gray or dimmed buttons mean that these frames and objectives have never been visited. These dimmed buttons can be turned to blue only after the corresponding objectives or frames are learned and evaluation results are updated to the database. Users can go to any other objective or frame by clicking the blue buttons. These buttons provide convenience for users to view any frames or objectives that have already been viewed without listening to the narrations. Figure 4.14 shows a screen with frame buttons.

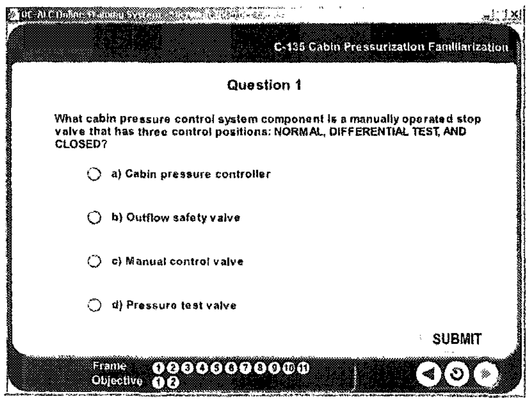

DC-AI C Online Training System
Team members when pneumatic carts used for pressurization
Exercise 3
Team members
Team lead
Second seat
Ground crew director
Pneumatic carts operator
Ground crew
Frame 12 13 14 15 16 17 18 19 20 21 22
Objective 1 2 3 4

Using your mouse, place each person icon in the proper location for an operational checkout using pneumatic carts for pressurization.

Figure 4.14: Typical lesson frame or screen with administrator and frame buttons

Figure 4.14 also illustrates that there is an additional set of small forward and backward buttons apart from the standard buttons available to users. These buttons are additional control features provided to support users with administrative privilege. As the administrator is not required to complete each objective frame by frame, he or she is provided with this feature to quickly browse through an objective's contents. This is a feature that does not appear for the normal user, as he or she must view all the frames completely and in order. The administrative buttons are separated from the primary navigation system and located at the left corner of the navigation bar. They are not integrated with the regular buttons because of their difference in functionality.

To evaluate and provide feedback to users and administrators, most objectives contain two questions or activities at the end that require action by the users. An example of a question frame is shown in Figure 4.15a. Normally questions are in the form of multiple-choices questions. These questions cover the basic concepts users should have learned throughout a particular objective. After an answer has been submitted, the system presents feedback to users that suggest the correct answer if the wrong answer was chosen or gives a congratulations message with a summary statement. Figure 4.15b shows an example of a feedback screen.

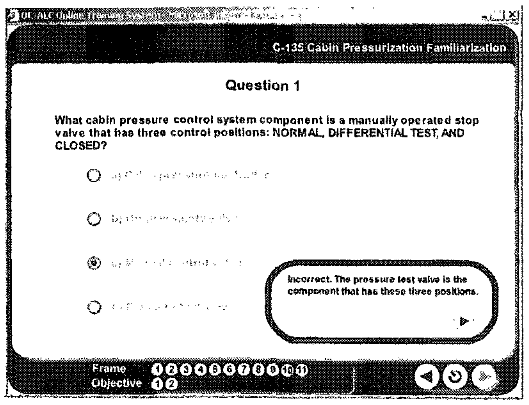


C-135 Cabin Pressurization Familiarization

Question 1

What cabin pressure control system component is a manually operated stop valve that has three control positions: NORMAL, DIFFERENTIAL TEST, AND CLOSED?

- a) Cabin pressure controller
- b) Outflow safety valve
- c) Manual control valve
- d) Pressure test valve



C-135 Cabin Pressurization Familiarization

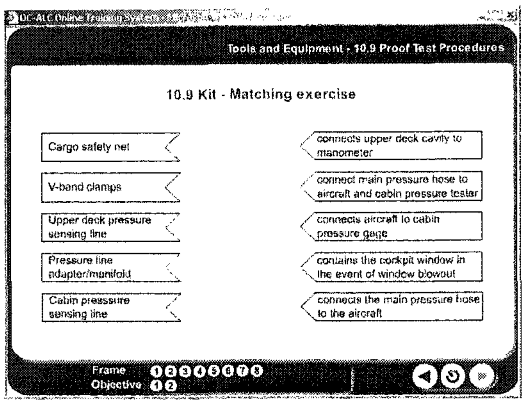
Question 1

What cabin pressure control system component is a manually operated stop valve that has three control positions: NORMAL, DIFFERENTIAL TEST, AND CLOSED?

Incorrect. The pressure test valve is the component that has these three positions.

Figures 4.15a and 4.15b: Typical evaluation question and question feedback frames

Besides the questions, there are various simulation activities throughout the lessons. These activities require users to participate by doing something, such as matching statements, solving puzzles, etc. A sample activity is shown in Figure 4.16.

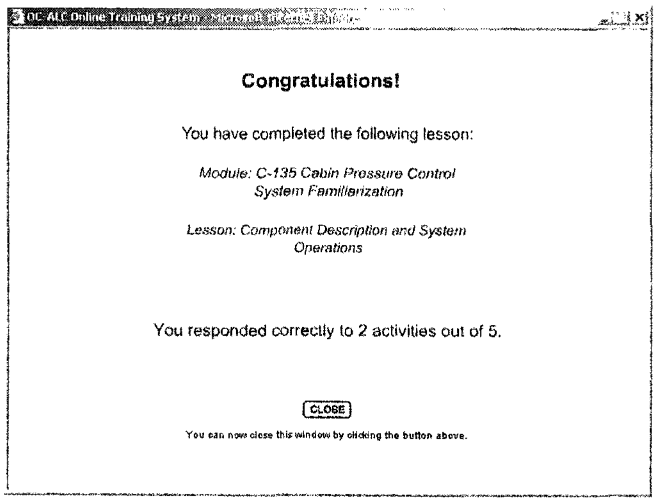

Tools and Equipment - 10.9 Proof Test Procedures

10.9 Kit - Matching exercise

Cargo safety net	connects upper deck cavity to manometer
V-band clamps	connect main pressure hose to aircraft and cabin pressure tester
Upper deck pressure sensing line	connects aircraft to cabin pressure gege
Pressure line adapter/manifold	contains the cockpit window in the event of window blowout
Cabin pressure sensing line	conneccs the main pressure hose to the aircraft

Figure 4.16: Example of activity (Matching Statements)

At the end of the lesson, the last screen will congratulate users for completing the lesson and also report how many questions or activities were completed successfully throughout the lesson. Figure 4.17 shows a congratulations page. Each user's progress can be viewed at any time by the progress link at the top menu bar in the main browser window. This feature will give all results for all lessons and objectives assigned. Figure 4.10 shows an example of this user progress screen.


Congratulations!

You have completed the following lesson:

Module: C-135 Cabin Pressure Control System Familiarization

Lesson: Component Description and System Operations

You responded correctly to 2 activities out of 5.

You can now close this window by clicking the button above.

Figure 4.17: Congratulations page

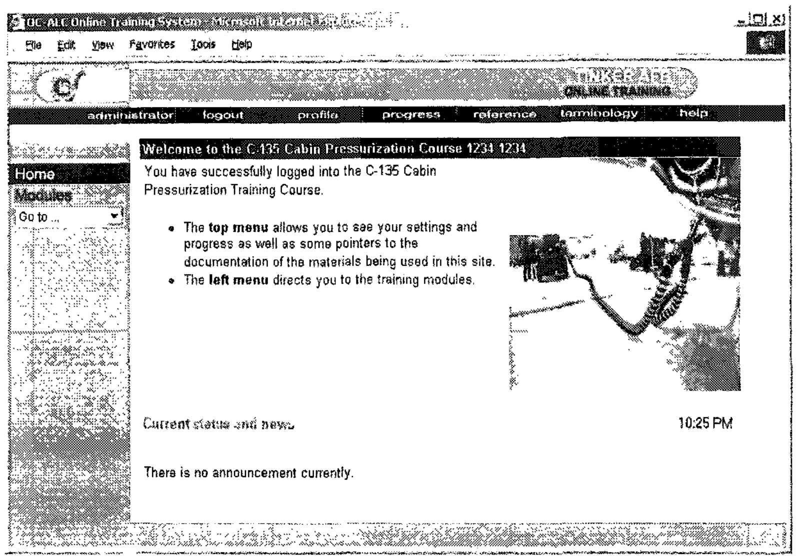
Server-based Database and User Tracking

To support the administrative section, all relevant user information is tracked and recorded in a database on the server of the web pages. The data tracking is done only at the completion of each objective and is then sent to the database. By accessing the online database only at the end of each objective, the frequency of interaction with the server is minimized. This results in a more responsive system when there are a large number of users. The information-tracking feature also provides users with the opportunity to leave at any point and to return exactly to the next incomplete objective.

Another functional use of the database is to regulate administrative privileges depending on the level of the user. Each user is classified into three levels: administrator, manager, and basic user. The difference between administrators and managers is that administrators have full control of the entire website, while managers are given only partial administrative capabilities to supervise small user groups.

Administrative System

A web-based administrative system is integrated to allow administrators to monitor users and to supervise the overall system through a web-based interface. This system facilitates only administrative user accounts to access, add, modify, delete, edit user information, set up user groups, manage lessons, and report user progress. Administrative functions are separated from the main training system and are accessed through a special link in the top menu bar that only appears for administrator and manager accounts as shown in Figure 4.18.



Welcome to the C-135 Cabin Pressurization Course 1234 1234

You have successfully logged into the C-135 Cabin Pressurization Training Course.

- The **top menu** allows you to see your settings and progress as well as some pointers to the documentation of the materials being used in this site.
- The **left menu** directs you to the training modules.

Current status and news

10:25 PM

There is no announcement currently.

Figure 4.18: A link to Administrator main page

Inside the administrator page, the left menu bar provides links to a number of options, such as modify profile information, system preferences, manage groups, manage departments, user progress, and guest user access information. Each individual feature is discussed in detail briefly as follows. This administrator page is illustrated in Figure 4.19.

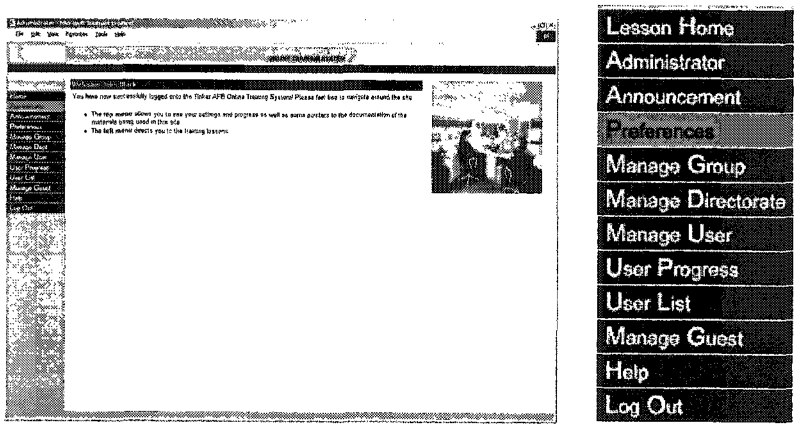
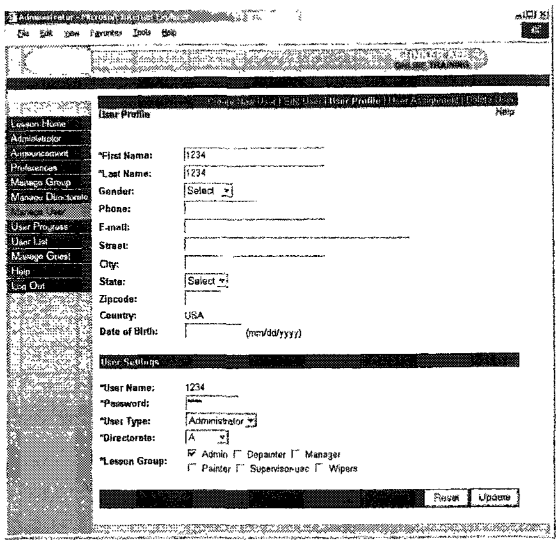


Figure 4.19: Administrator main page with maximize menu separately

(1) The *Administrator* is similar to the one available for basic users where personal information is displayed. Additional capability to modify personal information

and system settings is provided for administrators. An Administrator page is shown in Figure 4.20.



User Profile

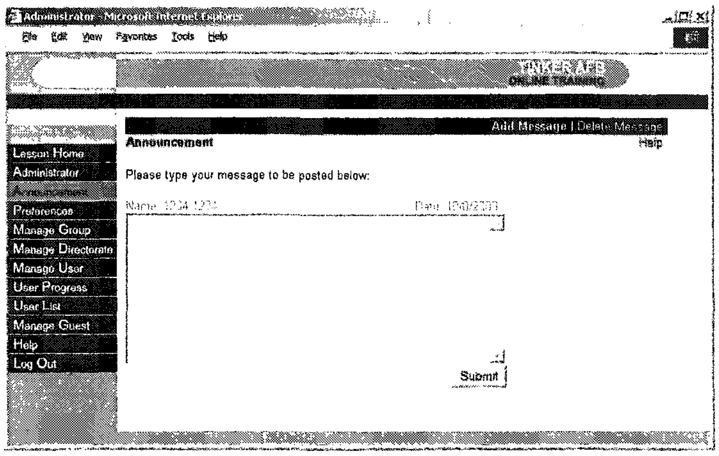
*First Name:	1234
*Last Name:	1234
Gender:	Select
Phone:	
E-mail:	
Street:	
City:	
State:	Select
Zipcode:	
Country:	USA
Date of Birth:	(mm/dd/yyyy)

User Settings

*User Name:	1234
*Password:	
*User Type:	Administrator
*Directorate:	A
*Lesson Group:	[✓ Admin] Depainter Manager Painter Supervisor-usc Wipers

Figure 4.20: Administrator page

(2) The *Announcement* allows the posting of announcements in the training system, and anyone who accesses the EM/C135 System welcome page can read the current news. The current news gives users general information about the system such as modified features and added contents. Figure 4.21 shows an *Announcement* page.



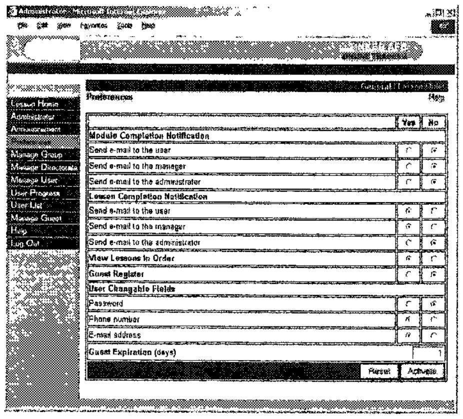
Announcement

Please type your message to be posted below:

Name: 1234 1234 Date: [illegible]

Figure 4.21: Announcement page

(3) The *Preferences* is a listing of features that administrators can use to change system attributes. It can be used to activate or deactivate preferences, such as email notification of lesson completion, modification of objectives' display order, and Guest account access, at any time. The sending email feature is implemented for the purpose of encouraging learners. The modification of objectives' display order option gives managers the opportunity to control the learning sequence. Guest account access, which is an important protection method, is controlled; the administrator can turn the guest registration on and off. This setting can block the non-Tinker user from entering the training system for security and traffic reasons. Another method to reduce the traffic is to minimize guest expiration days. The control of guests' registrations and expiration days is employed for reasons of both security and management. Some of the system options can be seen in Figure 4.22.



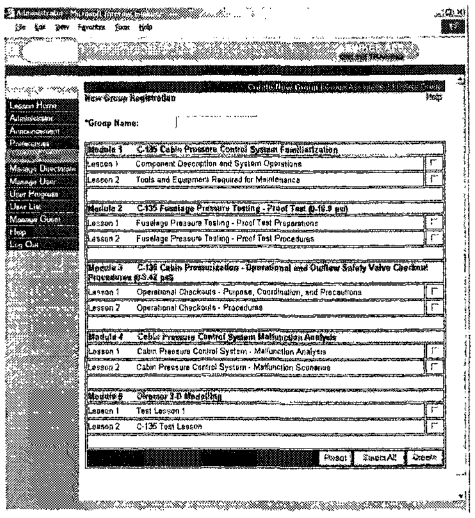
Preferences

	Yes	No
Module Completion Notification		
Send e-mail to the user	○	●
Send e-mail to the manager	○	●
Send e-mail to the administrator	○	●
Lesson Completion Notification		
Send e-mail to the user	●	○
Send e-mail to the manager	●	○
Send e-mail to the administrator	○	●
View Lessons In Order	●	○
Guest Register	○	●
User Changable Fields		
Password	○	●
Phone number	●	○
E-mail address	●	○
Guest Expiration (days)		1

Figure 4.22: Preference items

(4) The *Manage Group* section allows lesson assignment for the different groups of users accessing the system. A group can be set up in numerous ways, as the requirements vary based on the working background, such as wiper, painter, etc. Each

individual user can also be assigned additional lessons if needed. This feature allows the creation of multiple lesson groups. Each lesson group includes multiple lessons. Once a lesson group is created, it can be assigned to users. This avoids the situation of assigning multiple individual lessons to each user. A lesson group can be created by assigning a group a name, and then selecting lessons by checking the check box shown in Figure 4.23. An existing lesson group can be edited and deleted by clicking on Group Assignment and Delete Group buttons.



New Group Registration

*Group Name:

Module 1	C-135 Cabin Pressure Control System Familiarization	
Lesson 1	Component Description and System Operations	☐
Lesson 2	Tools and Equipment Required for Maintenance	☐
Module 2	C-135 Fuselage Pressure Testing - Proof Test (0-10.9 psi)	
Lesson 1	Fuselage Pressure Testing - Proof Test Preparations	☐
Lesson 2	Fuselage Pressure Testing - Proof Test Procedures	☐
Module 3	C-135 Cabin Pressurization - Operational and Outflow Safety Valve Checkout Procedures (0-3.42 psi)	
Lesson 1	Operational Checkouts - Purpose, Coordination, and Precautions	☐
Lesson 2	Operational Checkouts - Procedures	☐
Module 4	Cabin Pressure Control System Malfunction Analysis	
Lesson 1	Cabin Pressure Control System - Malfunction Analysis	☐
Lesson 2	Cabin Pressure Control System - Malfunction Scenarios	☐
Module 5	Director 3-D Modelling	
Lesson 1	Test Lesson 1	☐
Lesson 2	C-135 Test Lesson	☐

Figure 4.23: Manage group

(5) The *Manage Directorate* provides managerial users with the options to add, delete, and edit manager accounts. A manager account is like that of a basic user, except managers can add, delete, and edit users into his or her directorate. Each directorate has only one group attached to it so that the lessons can be defined. This feature is implemented to provide the opportunity of grouping users according to the directorate. The users in the same directorate can then be manipulated together. With this feature, a

directorate can be created by providing the information shown in Figure 4.24. Also a directorate can be edited and deleted.

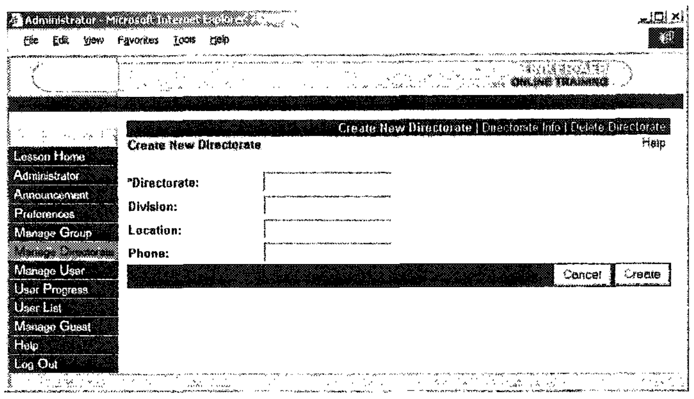
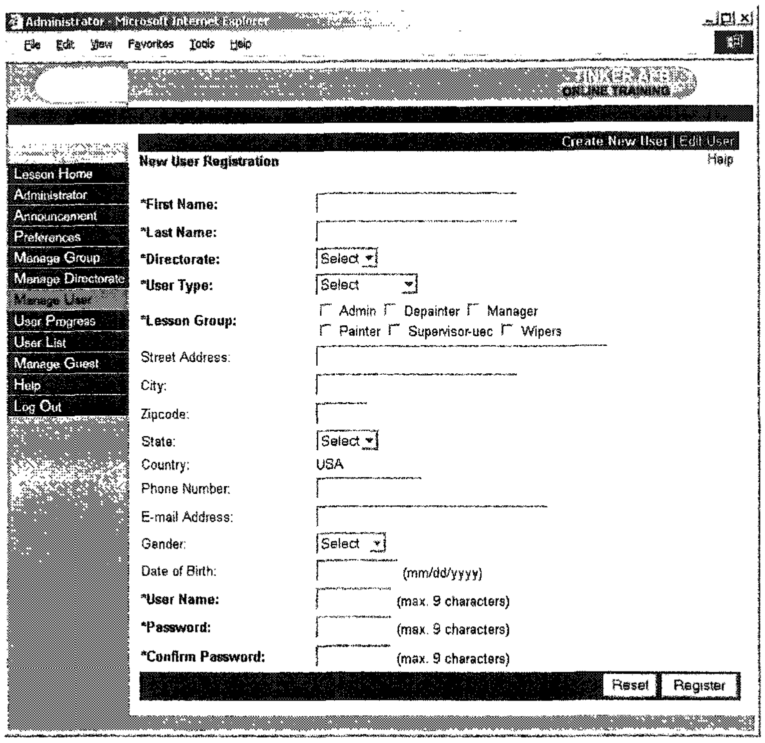


Figure 4.24: Manage directorate

(6) The *Manage User* supports the registration of new users into the system by entering required basic information that creates an identity through a New User Registration form as shown in Figure 4.25. This information allows users to be assigned by identifying them with a group and department. When a new user is created, directorate, user type, lesson group, user name, and password are assigned to the user. By assigning the user a user type, a user's control ability to this training system is decided. By assigning a single lesson group or multiple lesson groups to a user, the lessons that can be viewed by this user are determined. With a unique user name and password, a user can access the training. Since this feature is provided for the administrator instead of the normal user, the options like directorate, user type, and lesson group cannot be changed by the normal user himself. In order to make the administrators convenient to edit users, three methods are provided including name, lesson group, and directorate.

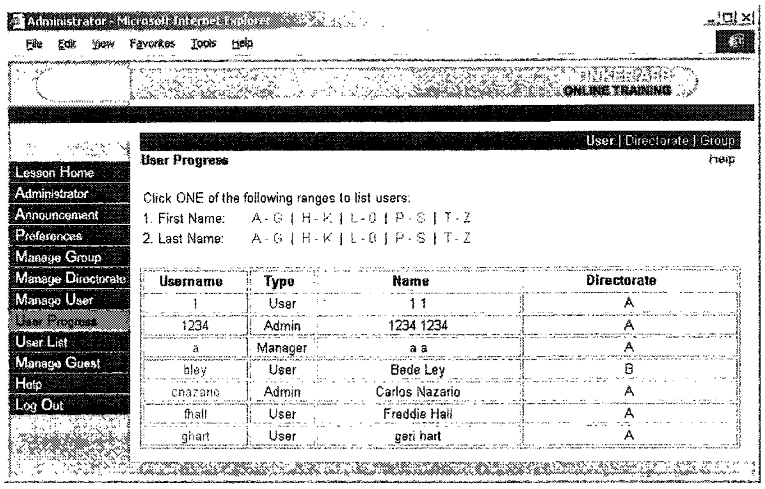


New User Registration

*First Name:	
*Last Name:	
*Directorate:	Select
*User Type:	Select
*Lesson Group:	Admin, Depainter, Manager, Painter, Supervisor-uec, Wipers
Street Address:	
City:	
Zipcode:	
State:	Select
Country:	USA
Phone Number:	
E-mail Address:	
Gender:	Select
Date of Birth:	(mm/dd/yyyy)
*User Name:	(max. 9 characters)
*Password:	(max. 9 characters)
*Confirm Password:	(max. 9 characters)

Figure 4.25: New User Registration form

(7) The *User Progress* is incorporated to provide administrative users with the option of previewing the performance based on individual users, groups, and departments. The progress information can be organized to support administrators in analyzing and evaluating the progress in a multi-faceted way. Figure 4.26 is a user progress list sorted by user's name. In this list, all users, including managers, normal users and guests, are listed. User progress lists can be sorted by Directorate and Group as well. If the administrator clicks on a user link, the corresponding user's progress page shown in Figure 4.12 appears. On the user's progress page, the administrator can gain users' performance and progress information such as: user names, the score, the time spent in each objective, last dates to access each objective, the completed objective (green color) and the uncompleted objective (blue color).



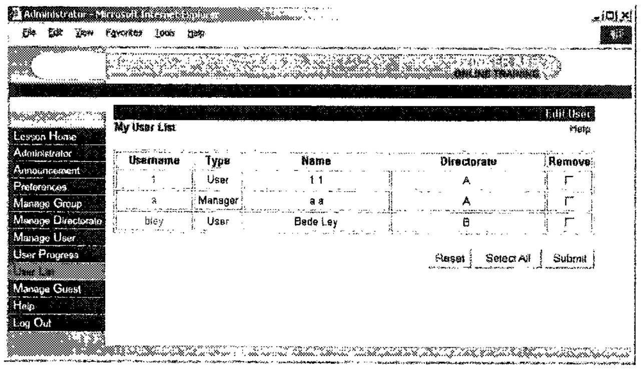
User Progress

Click ONE of the following ranges to list users:
1. First Name: A - G | H - K | L - O | P - S | T - Z
2. Last Name: A - G | H - K | L - O | P - S | T - Z

Username	Type	Name	Directorate
1	User	1 1	A
1234	Admin	1234 1234	A
a	Manager	a a	A
bley	User	Bede Ley	B
cnazario	Admin	Carlos Nazario	A
fhall	User	Freddie Hall	A
ghart	User	geri hart	A

Figure 4.26: User progress list

(8) The *User List* offers easy access to viewing or modifying user information. If administrators want to monitor several users, they can add them to the list. Once the users are added, they can be accessed directly in the future, without the necessity of finding them each time. Figure 4.27 shows a *User List* page.



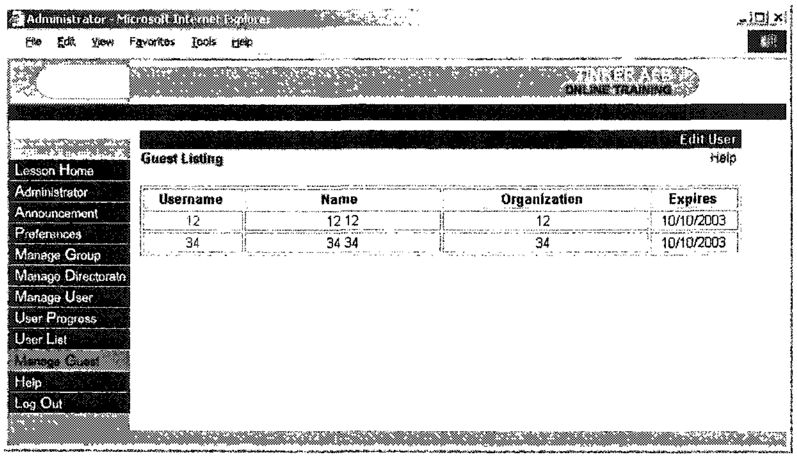
My User List

Username	Type	Name	Directorate	Remove
1	User	1 1	A	☐
a	Manager	a a	A	☐
bley	User	Bede Ley	B	☐

Figure 4.27: User List page

(9) The *Manage Guest* is designed to provide administrators with the option to decide what lesson content can be accessible to unauthorized guest users who enter the system as shown in Figure 4.28. This feature is the same as *Manage User* in feature capabilities such as editing guest information, viewing guest progress, and deleting

guests. However, it only applies to guest accounts, and this type of account does not appear on any progress report and would not interfere with Tinker user account type. Additionally, this feature also provides the option to alter a guest account to user account.

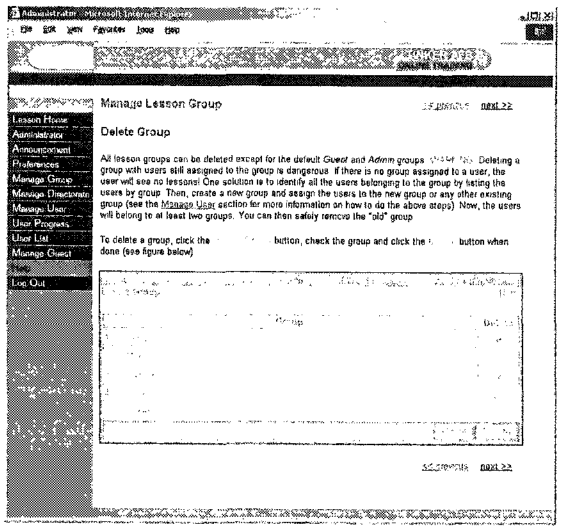


Guest Listing

Username	Name	Organization	Expires
12	12 12	12	10/10/2003
34	34 34	34	10/10/2003

Figure 4.28: Manage Guest page

(10) The *Help* assists administrators in mastering management tools so that the management process is easy and fast. This is the final feature available for administrators and managers. Figure 4.29 illustrates the *Help* page.



Manage Lesson Group

Delete Group

All lesson groups can be deleted except for the default *Guest* and *Admin* groups. Deleting a group with users still assigned to the group is dangerous. If there is no group assigned to a user, the user will see no lessons! One solution is to identify all the users belonging to the group by listing the users by group. Then, create a new group and assign the users to the new group or any other existing group (see the Manage User section for more information on how to do the above steps). Now, the users will belong to at least two groups. You can then safely remove the "old" group

To delete a group, click the button, check the group and click the button when done (see figure below)

Figure 4.29: Help page

CHAPTER 5: THE DEVELOPMENT OF CSD AND 3D DYNAMICS SYSTEM

As mentioned in the previously reviewed web-based learning and training systems, currently, several learning and training systems for both educational and industrial purposes incorporate a variety of electronic media over the Internet. Three-dimensional visualization has played a major role in existing technical learning environments. However, the method of implementing three-dimensional visualization can still be improved in order to deliver the content more conveniently. Therefore, one of the major goals of this research is to identify and illustrate the implementation of new techniques to provide an improved environment.

This chapter presents the various technical aspects involved in designing a web-based training system. The different components that constitute a training system are learning environment and front-end system designs. In this chapter, these two components are discussed. Learning environment design is divided into learning system structure, content, and feature, while front-end system design is comprised of user interface design and navigation system design. However, for the design of the 3D Dynamics System, only system content design is discussed because the system is an extended topic of the EM/C135 System, which was discussed in the previous chapter.

CSD System

1. Learning Environment Design

1.1 Learning System Structure

In order to manage and effectively organize the materials, a linear structure is incorporated in designing the organization of the learning system. The structure is designed in close relationship to that of an academic curriculum. Each level can be compared to a traditional education system. Three sections are organized in sequence to maintain a linear learning process. In addition, the learning system structure for this particular demonstration is ordered in a way that complies with the requirement of the Tinker Air Force Base (AFB) project description. These three successive sections are considered three levels of learning as well; they are integrated into the learning system to linearly maintain the structure contents that provide the user an opportunity to gain the most information out of the learning materials. These sections include *Part Breakdown*, or overview or introduction level, the *Assembly Process Tutorial*, or the material content learning level, and the *Assembly Process Evaluation* section, which is also considered the material content assessment level. Figure 5.1 illustrates the learning structure of the CSD System.

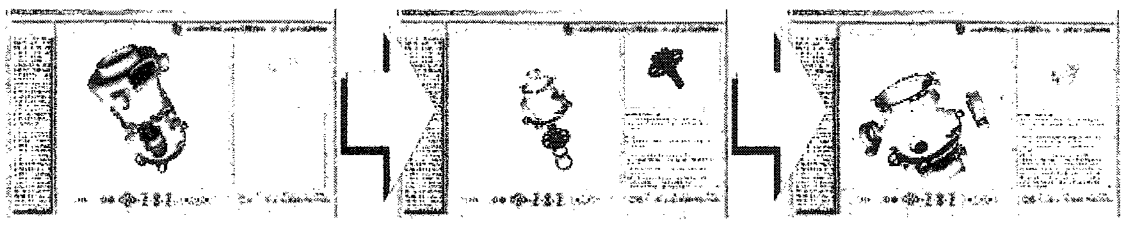


Figure 5.1: Learning structure of the CSD System

This style of learning system is equivalent to the learning system commonly presented in the traditional education system, where a learner has to study the overview or introduction of a topic first, and then proceed to the learning materials in detail, and finally is required to take an examination to evaluate the level of understanding the course materials. Inside each section, the learning material is classified into frames or steps in order to make the content easier to understand and to make the most logical and effective use of the learning process. Each frame displays specific content, and is made self-contained in a single environment consecutively in order to gradually introduce the learning materials for a complete single learning module that represents an entire specific topic, such as *Aircraft Part Maintenance and Assembly Process*. This allows a user to learn the material in pieces from the beginning to the closing stages step-by-step and then move on to the next module or assembly part if available. An organization of complete frames or steps included in the CSD System is shown in Figure 5.2.

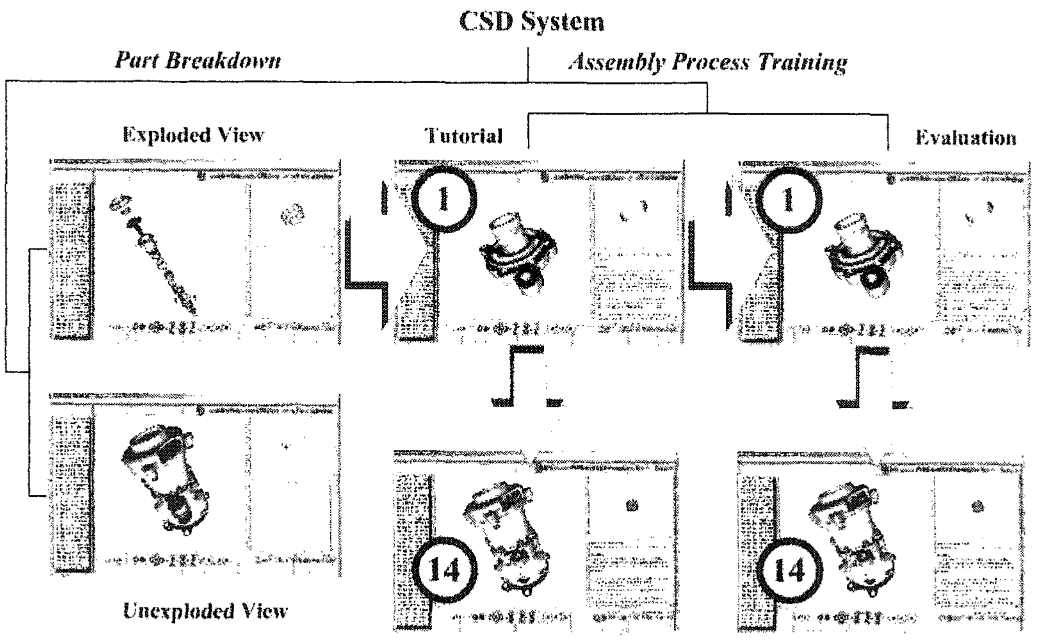


Figure 5.2: An organization of complete frames or steps included in the CSD System

1.2 Covered Topics

To illustrate the effectiveness of three-dimensional supportive capability in the Shockwave 3D technology primarily employed in this research, a demonstration environment is required. The demonstration environment serves as a complete training system for a specific subject. The material contents that have been chosen for this research to be presented in the demonstration environment focus on the topic of *Aircraft Part Maintenance and Assembly Process*. This topic is excellent for a demonstration because it requires the assistance of three-dimensional visualization to describe the concepts more productively. Moreover, this topic covers all details required in the project description funded by Tinker AFB. This system is not only a demonstration environment for this research but it is also part of a demonstration environment for Tinker AFB in a project to launch a complete web-based training system for the topic of *Aircraft Part Maintenance and Assembly Process*. Tinker AFB plans to use this environment for its personnel to study the possibility of developing a complete system in the future.

1.3 System Purpose and Introduction

The main reason for Tinker AFB to have this web-based environment developed is to provide a flexible and effective training system that is accessible as a resourceful, stimulating, attractive, and challenging training method to serve a number of personnel simultaneously and to catch the attention of personnel, which is the utmost concern for Tinker AFB. These are some of the main benefits of using an interactive web-based training system instead of traditional training techniques. The main objective of the CSD System is to guide and instruct Tinker AFB personnel to correctly perform specific *Aircraft Part Maintenance and Assembly Process* procedure. The system consists of three major sections, including *Part Breakdown*, *Assembly Process Tutorial*, and the *Assembly Process Evaluation* section.

The *Constant Speed Mechanical Drive (CSMD) Mechanism* is chosen by Tinker AFB as a specific part for the demonstration environment. The *CSMD Mechanism* is manufactured by Sundstrand Aviation, a unit of the Sundstrand Corporation, located in Rockford, Illinois. The *CSMD Mechanism* is illustrated in Figure 5.3. The purpose of the *CSMD Mechanism* is to drive each generator used in an aircraft. Normally each aircraft employs two generators. The generators are the primary source of electrical power for the aircraft. The *CSMD Mechanism* makes it possible to obtain alternating current with a stable frequency, even though the source of rotary power for the *CSMD Mechanism* from aircraft engines operates at various speeds. The *CSMD Mechanism* converts varying input speeds into a constant output speed as required. This constant output speed drives a generator, which then produces output power.

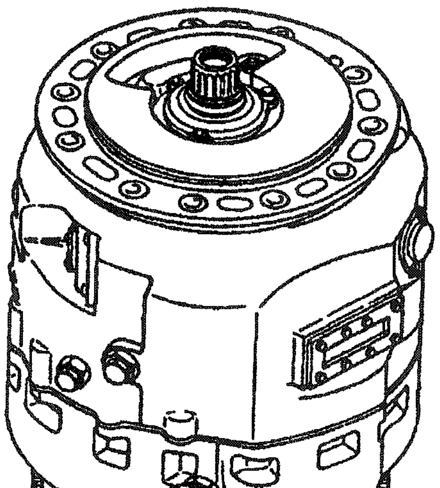


Figure 5.3: The CSMD Mechanism

The *CSMD Mechanism* consists of a number of subassemblies; they are the *Left Hand Pump and Motor Assembly*, *Right Hand Pump and Motor Assembly*, *Governor*, and *Differential*. However, only the *Left Hand Pump and Motor (LHPM) Assembly* as illustrated in Figure 5.4 is addressed in this demonstration because this subassembly is the main component in the *CSMD Mechanism* and the most complex subassembly as well. Therefore, if this subassembly part is completed first as a prototype, then the other parts in the *CSMD Mechanism* can be developed straightforwardly following the same approach of development with less complexity.

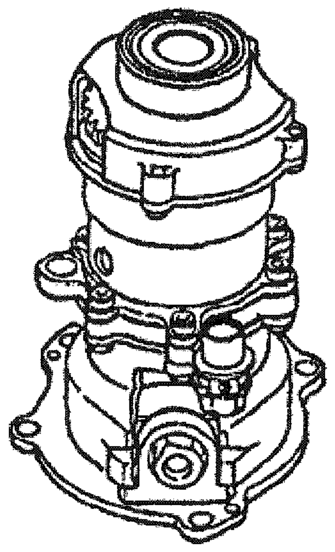


Figure 5.4: The LHPM Assembly

Several steps are required to completely and properly assemble the *LHPM Assembly*. A step-by-step procedure must be followed in order to obtain a correct assembly. The *Assembly Process Tutorial* and *Assembly Process Evaluation* sections are identically divided into frames that refer to each step defined in the Technical Order (TO) that is provided by Tinker AFB. The content covered in the system is the same as the materials contained in an official TO but with additional advantages, such as user interaction with and convenient accessibility to the training materials. The main difference between these two sections is that the *Assembly Process Tutorial* section demonstrates how to perform the assembly principally using animation techniques with text instruction description, and the *Assembly Process Evaluation* section allows users to assemble the *LHPM Assembly* on their own, through the use of simulation techniques, using the knowledge they have learned in the *Assembly Process Tutorial* section. This is to support the Tinker AFB requirements as presented in the TO, and to increase the learning progress of users as well. There are fourteen frames or steps in the *Assembly Process Tutorial* section, some of which involve multiple subparts as precisely staged in the TO, each of which represents one step in the step-by-step procedure involved in the assembly of the *LHPM Assembly*.

1.4 Learning System Features

1.4.1 First Section: *Part Breakdown* section

The beginning level or the first section in the system structure, *Part Breakdown*, is divided into two steps; the first step shows an exploded view of the assembly and the second step shows an unexploded view of the assembly. This stage is similar to an overview in academic curriculum, where a student has an opportunity to obtain the basic knowledge involved in a specific assembly. This section provides users with a general overview of assembly background information, including an overall look at the complete assembly in both exploded and unexploded views, how many identical parts are included in an assembly, part location, and part names as well. This information is considered to be important supplementary information needed to extend users' ability to properly complete the assembly process. This fundamental background information assists users in continuing and completing the next two sections with a full understanding of the materials.

1.4.2 Second Section: *Assembly Process Tutorial* section

The purpose of the second section is to teach users how to assemble specific assembly parts, and is designed to closely mimic the real specific part assembly process. In the second level or section of the system, users learn how to assemble a particular assembly step-by-step from the beginning until the assembly is completed by using animation technique through three-dimensional models along with text instruction description. The content of this section is sequentially organized, which follows an official TO. In order to provide users with flexibility during learning, the tutorial consists of a number of frames, each of which represents a step in the assembly process for the user to accomplish in order to follow the official TO.

1.4.3 Third Section: *Assembly Process Evaluation* section

To ensure that users clearly understand the assembly procedure, a training evaluation of the assembly process performance for the specific aircraft part (the *LHPM Assembly*) is required. Upon completion of the tutorial section regarding the knowledge of assembly process, the user has an opportunity to test his or her comprehension on the material and to collect information about how well he or she understands the content. This learning method places the most demand upon users in terms of interactivity, especially as compared to the first two sections. It provides actual examples of the procedures users have to perform in a particular time period, as compared to the previous section, in which users merely viewed the procedures. This learning method deepens users' understanding of the material taught in the previous section, and serves as a self-assessment for a user to realize how much knowledge he or she has gained throughout the training process. The *Assembly Process Evaluation* section is intensely involved with simulation that allows users to learn engineering and technical concepts through investigation and exploration. In this section, users have to perform exactly a step-by-step assembly procedure as demonstrated in the second section through a simulation. The simulation allows users to complete an assembly, part by part, by moving from the default location to the correct one. This section concludes with the complete assembly part as illustrated in the information section.

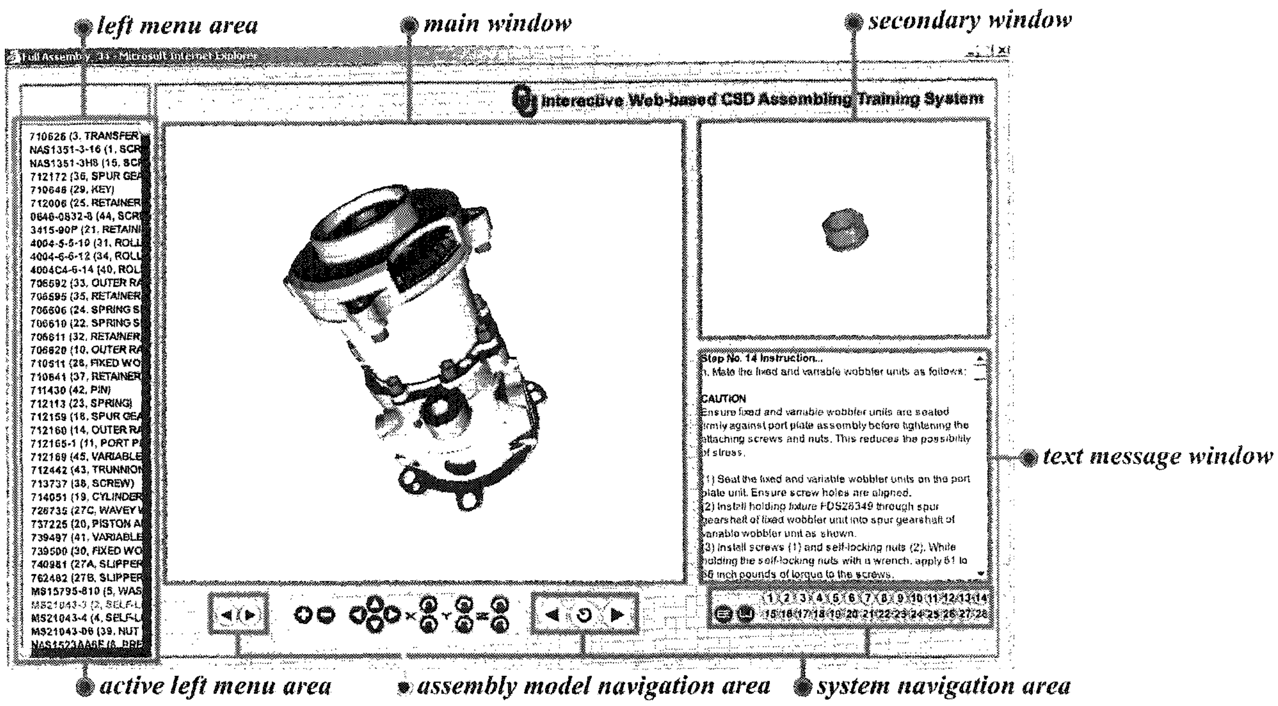
In the simulation, users have to move the models involved in a particular step to their correct position in the assembly and in the right order of assembly. A user has to perform exactly as though he or she was completing a real assembly using real models. However, instead of completing a real assembly right away, he or she has to complete an assembly through computer simulation first. This is the last step in the system to assess a user's level of understanding.

As the structure of this system is constructed and maintained from the first section to the last section, as explained, this system provides a fundamental learning experience for students using explicit material that focus on specific topics. This complete demonstration system is considered a building block of the overall system structure, and is comparable to an entire chapter in a textbook. To a user, it appears as if he or she is working in an actual environment with the real assembly part, although with the added benefits of interactive instruction and message cues. Having learned the material, a user may then be able to properly complete the assembly.

2. Front-end System Design

The interface of a website is one of the first factors users notice and, therefore, it is important to create an attractive visual appearance while keeping it simple. The front-end system design consists of the arrangement and design of the overall interface, navigation buttons, and material presentation like text, graphics, animations, and simulations. Attractive and consistent visual appearance template, easy navigation, and lively content explanations are the goals of front-end system design in order to distinguish the system website from the rest.

The purpose of front-end layout design is to make the system flexible to navigate, easy to read, and consistent. The system layout template is formed to facilitate the front-end system design and to maintain the consistency of the presentation throughout the system. The template design is illustrated in Figure 5.5. The frame is composed of a training material presentation area and the navigation area.



Step No. 14 Instruction...
h. Mate the fixed and variable wobbler units as follows:

CAUTION
Ensure fixed and variable wobbler units are seated firmly against port plate assembly before tightening the attaching screws and nuts. This reduces the possibility of stress.

(1) Seat the fixed and variable wobbler units on the port plate unit. Ensure screw holes are aligned.
(2) Install holding fixture FDS28349 through spur gearshaft of fixed wobbler unit into spur gearshaft of variable wobbler unit as shown.
(3) Install screws (1) and self-locking nuts (2). While holding the self-locking nuts with a wrench, apply 61 to 66 inch pounds of torque to the screws.

Figure 5.5: Template design of the CSD System

The training material presentation area is the place used to illustrate the training content, which are explained with a variety of multimedia technologies such as simulation, animation, graphics, and text. The effectiveness of the web training is directly related to the level of system interactivity, which can capture learners' attention and stimulate their learning interest. A training material presentation area consists of the *main window*, *secondary window*, *text message window*, and *left menu area*. The navigation area is where users are supplied with learning system navigation and three-dimensional assembly model control features. The navigation area is discussed in more detail later in this chapter. A navigation area is composed of a *system navigation area*, *assembly model navigation area*, and *active left menu area* that is only available in the *Part Breakdown* section.

2.1 User Interface Design

As mentioned in the previous section, the training material presentation area consists of the *main window*, *secondary window*, *text message window*, and *left menu area*. The *main window* is the main area for the presentation; it is where the three-dimensional assembly model is presented as the major component used to explain the concepts for this learning system. It occupies the largest area in the system layout, as shown in Figure 5.5, due in part because the *main window* is the most important area for the learning system. The *main window* is where all step-by-step assembly process animations and simulations are correspondingly displayed for the *Assembly Process Tutorial* and *Assembly Process Evaluation* sections and exploded and unexploded views of the three-dimensional assembly model are illustrated for the *Part Breakdown* section. The three-dimensional assembly model used in the *main window* is a complete, single three-dimensional model file created using CAD software (Pro/ENGINEER), and then converted to *w3d* format, which is the only three-dimensional model format supported in Shockwave 3D (Director 3D). Moreover, throughout the learning process in the CSD System, two modifications are available for the three-dimensional assembly model: translation and transparent appearance that are controlled by the function only.

The *secondary window* is another area that displays the three-dimensional model but only each individual subpart model, not an entire assembly model. The size of this window is much smaller than the *main window*, as shown in Figure 5.5, because only a single three-dimensional part model appears in this window at a time. This window is used to display a particular part that has been selected in the *main window* while users are in the *Part Breakdown* section. In addition, the *secondary window* shows a specific part that is animated in that period of time during the assembly process animations when users are in the *Assembly Process Tutorial* section. Similarly, as users are in the *Assembly Process Evaluation* section, the *secondary window* demonstrates a specific simulated part in that period of time through the assembly process simulations. The three-dimensional models for subparts used in this window are in separate three-dimensional model files, and each subpart has own file. They are created using CAD software (Pro/ENGINEER) as well, and then converted to *w3d* format.

Each individual subpart in the three-dimensional assembly model in the *main window* and each individual three-dimensional model for each subpart used in the *secondary window* are distinguished by colors, including *red*, *green*, *blue*, and *yellow*, assigned to different subparts. The color of each part is identical when displayed in the *main window* and the *secondary window* in order to ensure that the part viewed in one window is the same as that viewed in the other. Since there are so many subparts involved in the three-dimensional assembly model, some of them have been assigned the same color. However, the locations of the parts that share a color are not adjacent to each other so that subparts can be differentiated effectively. Figure 5.6 shows sample subparts that are assigned with different colors. Figure 5.7 shows sample subparts that are assigned with the same color but are not alongside each other.

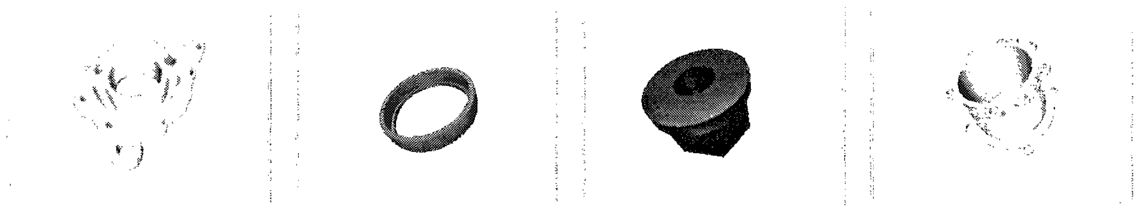


Figure 5.6: Sample subparts that are assigned with different colors

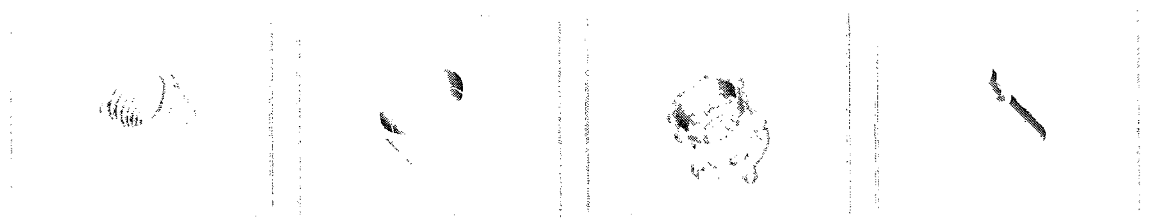


Figure 5.7: Sample same color subparts that are not alongside each other

The *left menu area*, as illustrated in Figure 5.5, contains the part name and part Identification Number (ID) for all subparts contained in the assembly. Because there are several parts involved in the assembly, the menu is provided with both horizontal and vertical scroll bars that can move up, down, left, and right to allow users to get complete information for each part. To implement these scroll bars, the position of the scroll bar buttons are the keys used to control the position of all part names in order to make them act as scroll bars that can move all of the information included in the *left menu area* simultaneously.

The top, bottom, left, and right boundaries of the *left menu area* are in a fixed position horizontally and vertically as part of the implementation so users cannot view any information outside of the boundaries. Actually, only the scroll bar buttons are movable, part names are not. When scroll bar buttons are moved by users, the position of the scroll bar buttons will be changed, and then the difference between the original and current positions of the scroll bar buttons is calculated. This difference in distance is used to control the position of each text sprite that represents part name included in the *left menu area*. To control the position of each text sprite, the position difference of the scroll bar buttons is added to the original position of each text sprite to make each of them move smoothly along with the scroll bar buttons. This mechanism is implemented using Lingo scripting language. The only part name modification that is controlled by function associates with each part name itself is the color change from black to red when the part name is selected. This is also implemented using Lingo. The *left menu area* is positioned on the left side of the system layout and it covers from the top to the bottom of the system layout as well. Figure 5.8 illustrates key features, such as scroll bars, scroll bar buttons, and boundaries, that have been mentioned recently for the implementation of the *left menu area*.

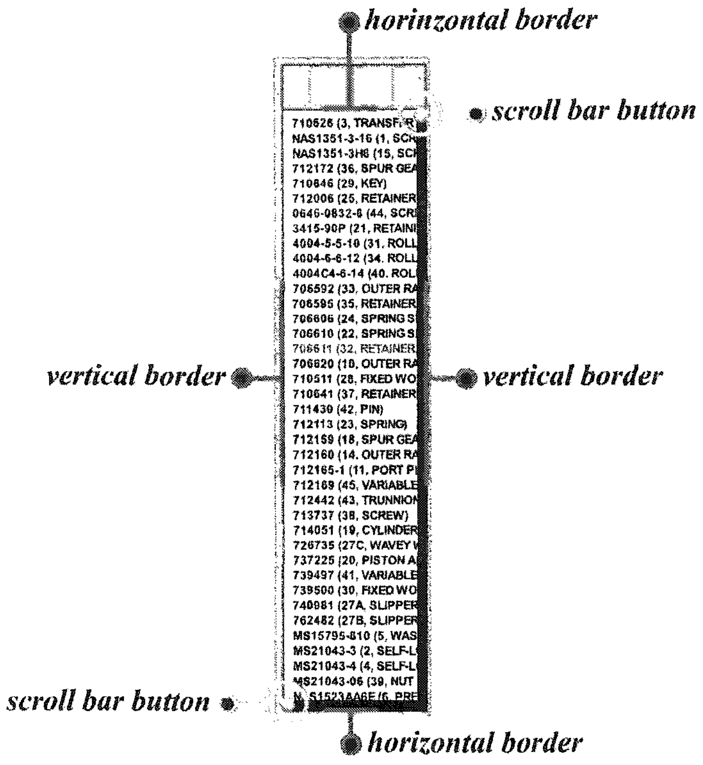


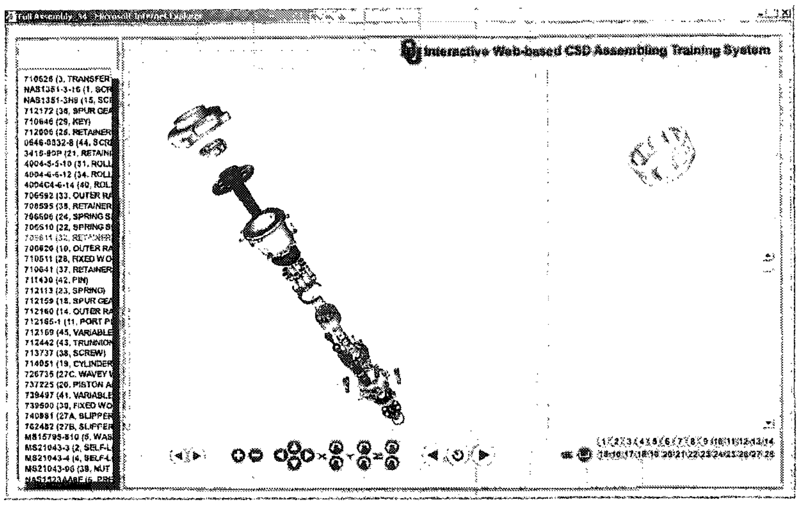
Figure 5.8: Key features for the implementation of the *left menu area*

The last element in the material presentation area is the *text message window* (Figure 5.5). This area is used to display assembly procedures in text as exactly presented in the official TO. The text instruction description appears alongside with assembly process animation and simulation in the *Assembly Process Tutorial* and *Assembly Process Evaluation* sections. The *text message window* also comes with vertical scroll bar when the text instruction description occupies more area than the size of *text message window* itself. The location of this window is on the right side of the system layout, below the *secondary window*. In addition, the size of this window is slightly larger than the size of the *secondary window*.

2.1.1 First Section: *Part Breakdown* section

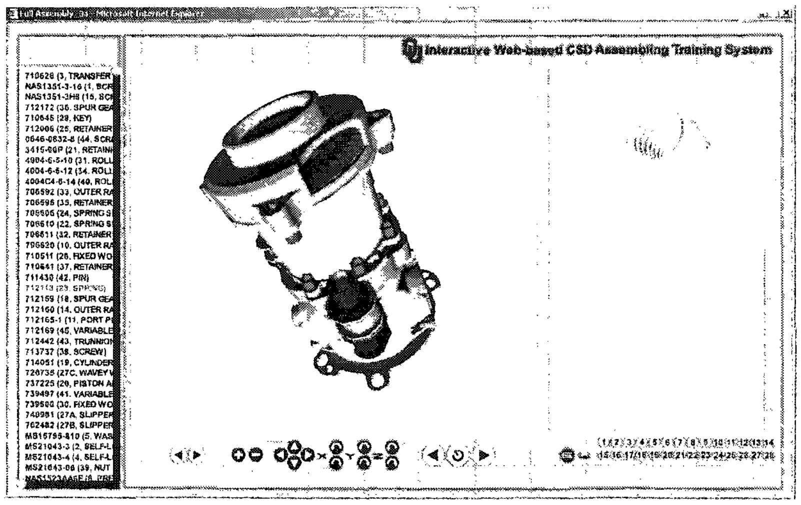
As mentioned, three successive sections are organized in order to provide a linear learning process so that users can acquire most of the information contained in the learning materials. The *Part Breakdown* section is the first section that gives users a general idea of the training content. The information included in this section is necessary for users to proceed to subsequent sections and to successfully understand the concepts presented. The three-dimensional models presented in this section are the main element used to provide users with information such as what subparts look like and where they are located in the assembly. This is beneficial for users because users can actually observe three-dimensional models and then obtain necessary information directly from what they perceive.

In the first section, the *main window* displays the complete assembly part in both exploded and unexploded views in order for users to explore what subparts are included in the assembly along with information such as part name, part ID, part location, and physical appearance of the part. Additionally, an exploded view can give users an idea of the subparts inside the assembly that cannot be seen from an unexploded view. An unexploded view is represented by the same three-dimensional model as an exploded view but the location of each subpart is repositioned to breakdown the assembly. Lingo scripting language is used to control the position of the subparts. Example screens of *Part Breakdown* are illustrated in Figures 5.9a and 5.9b.



710626 (3, TRANSFER
NAS1351-3-16 (1, SCR
NAS1351-3H8 (15, SCI
712172 (36, SPUR GEA
710646 (29, KEY)
712006 (26, RETAINER
0646-0832-8 (44, SCR
3415-80P (21, RETAINI
4004-5-5-10 (31, ROLL
4004-6-6-12 (34, ROLL
4004C4-6-14 (40, ROL
706592 (33, OUTER RA
706595 (35, RETAINER
706606 (24, SPRING S
706610 (22, SPRING S
706611 (32, RETAINER
706626 (10, OUTER RA
710511 (28, FIXED WO
710641 (37, RETAINER
711430 (42, PIN)
712113 (23, SPRING)
712159 (18, SPUR GEA
712160 (14, OUTER RA
712165-1 (11, PORT P
712169 (45, VARIABLE
712442 (43, TRUNNION
713737 (38, SCREW)
714051 (19, CYLINDER
726735 (27C, WAVEY W
737225 (20, PISTON A
739497 (41, VARIABLE
739500 (30, FIXED WO
740981 (27A, SLIPPER
762482 (27B, SLIPPER
MS15795-810 (5, WAS
MS21043-3 (2, SELF-L
MS21043-4 (4, SELF-L
MS21043-06 (39, NUT
NAS1523AA8E (6, PRE

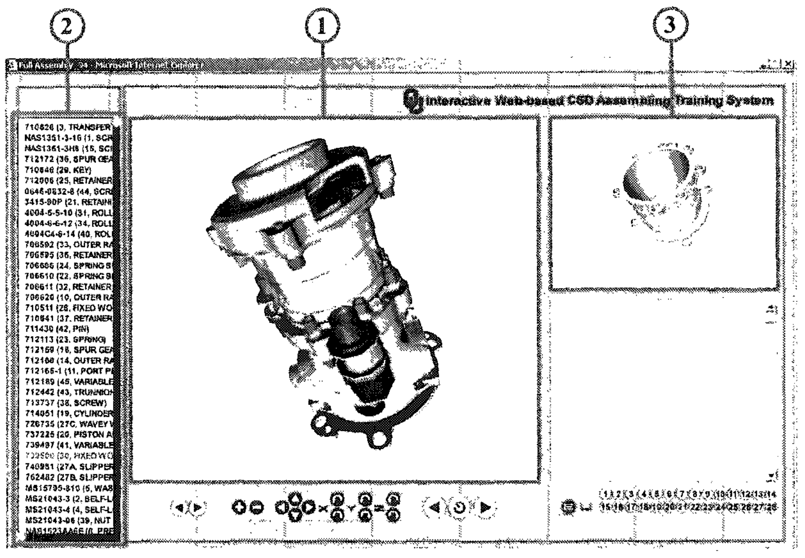
Figure 5.9a: Example of a screen in the *Part Breakdown* section: exploded view



710626 (3, TRANSFER
NAS1351-3-16 (1, SCR
NAS1351-3H8 (15, SCI
712172 (36, SPUR GEA
710646 (29, KEY)
712006 (26, RETAINER
0646-0832-8 (44, SCR
3415-80P (21, RETAINI
4004-5-5-10 (31, ROLL
4004-6-6-12 (34, ROLL
4004C4-6-14 (40, ROL
706592 (33, OUTER RA
706595 (35, RETAINER
706606 (24, SPRING S
706610 (22, SPRING S
706611 (32, RETAINER
706626 (10, OUTER RA
710511 (28, FIXED WO
710641 (37, RETAINER
711430 (42, PIN)
712113 (23, SPRING)
712159 (18, SPUR GEA
712160 (14, OUTER RA
712165-1 (11, PORT P
712169 (45, VARIABLE
712442 (43, TRUNNION
713737 (38, SCREW)
714051 (19, CYLINDER
726735 (27C, WAVEY W
737225 (20, PISTON A
739497 (41, VARIABLE
739500 (30, FIXED WO
740981 (27A, SLIPPER
762482 (27B, SLIPPER
MS15795-810 (5, WAS
MS21043-3 (2, SELF-L
MS21043-4 (4, SELF-L
MS21043-06 (39, NUT
NAS1523AA8E (6, PRE

Figure 5.9b: Example of a screen in the *Part Breakdown* section: unexploded view

In this section, two main interactive functions are available for users to explore the information contained in the *Part Breakdown* section. The first function is the *selected assembly subpart* function, this function executes anytime users select an individual subpart of the assembly in the *main window*. Users are able to select each part in order to learn information about that particular part. The part will become transparent after it has been selected, which helps users distinguish a particular part from the rest of the assembly parts. Only the selected part appears in the *secondary window* as well. This provides an opportunity for users to discover a particular part in more detail. The part can then be rotated using the input device, such as a mouse and touch pad, so that users can see what it looks like from all angles. As a part is selected, the color of the part name in the *left menu area* changes as well in order to give users that particular part name and part ID. Figure 5.10 shows the information displayed in a single execution of the *selected assembly subpart* function.



2
1
3
Full Assembly 34 - Microsoft Internet Explorer
Interactive Web-based CSD Assembling Training System

710626 (3, TRANSFER
NAS1351-3-16 (1, SCR
NAS1351-3H8 (15, SC
712172 (36, SPUR GEA
710646 (29, KEY)
712006 (25, RETAINER
0646-0832-8 (44, SCR
3415-90P (21, RETAIN
4004-6-5-10 (31, ROLL
4004-6-6-12 (34, ROLL
4004C4-6-14 (40, ROL
706592 (33, OUTER RA
706595 (36, RETAINER
706606 (24, SPRING S
706610 (22, SPRING S
706611 (32, RETAINER
706626 (10, OUTER RA
710511 (26, FIXED WO
710641 (37, RETAINER
711430 (42, PIN)
712113 (23, SPRING)
712159 (16, SPUR GEA
712160 (14, OUTER RA
712166-1 (11, PORT P
712189 (45, VARIABLE
712442 (43, TRUNNIO
713737 (38, SCREW)
714051 (19, CYLINDER
726735 (27C, WAVEY W
737225 (20, PISTON A
739497 (41, VARIABLE
739500 (30, FIXED WO
740951 (27A, SLIPPER
762482 (27B, SLIPPER
MS15795-810 (5, WAS
MS21043-3 (2, SELF-L
MS21043-4 (4, SELF-L
MS21043-06 (39, NUT
NAS1523AA6E (6, PRE

Figure 5.10: Information displayed in *selected assembly subpart* function execution

The flowchart that illustrates the mechanism of the *selected assembly subpart* function IS shown in figure 5.11. The implementation of this function mechanism initiates when users select a subpart in the assembly. Then, the first Lingo scripting language function detects if there is any model under the input device location, such as a mouse and notepad, and identifies that model. This function then sends a model name as a parameter to the functions that perform other operations, such as making that particular part in the three-dimensional assembly model transparent, displaying an appropriate three-dimensional model in the *secondary window*, or highlighting the part name and part ID in the *left menu area*. If no model is found then no function executes. This *model detection* function (*selected assembly subpart* function) is associated only in the *Part Breakdown* section.

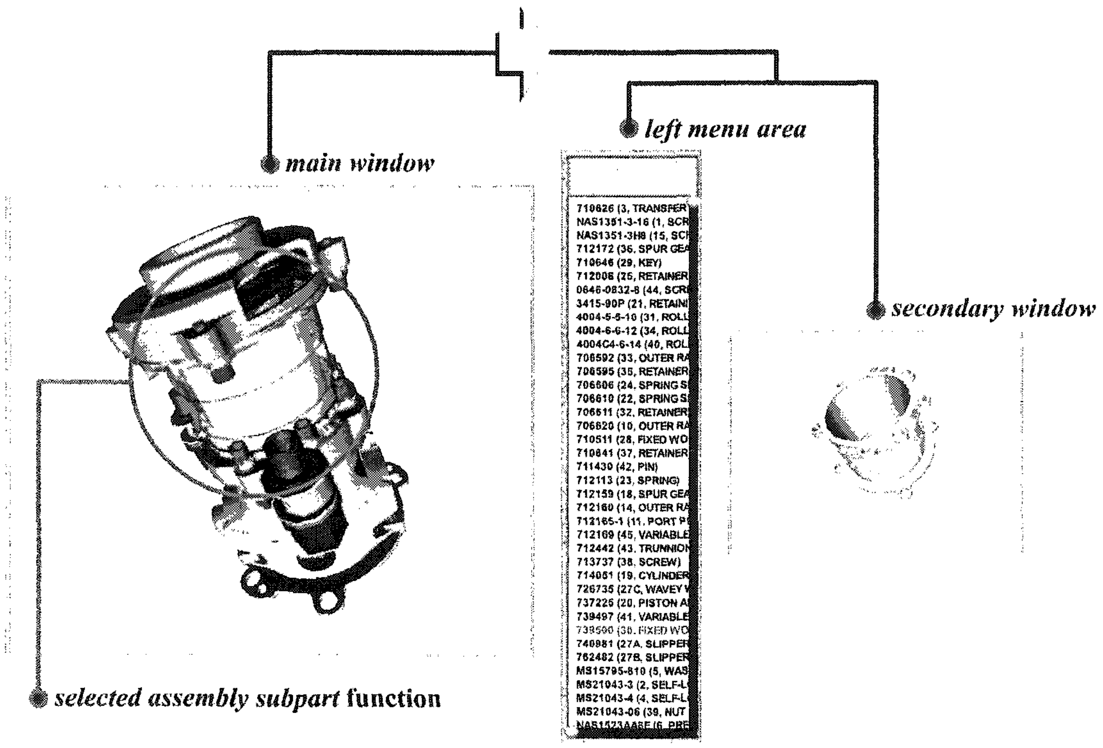


Figure 5.11: Flowchart of the *selected assembly subpart* function in *Part Breakdown*

The second function offered in the *Part Breakdown* section is integrated as an additional feature that can be activated only in this section. It is a *selected part name* function associated in the active *left menu area* that contains all part names and part IDs included in this particular assembly part (the *LHPM Assembly*). Users can click on any part name in order to view what that specific part looks like and where it is located in the assembly part. As a part name is selected, the color of the part name changes and the part in the *main window* becomes transparent along with the appearance of that part in the *secondary window*. This feature helps users better understand the basic information for a particular part and also the assembly part based on the part name instead of the part physical appearance used to initiate the *selected assembly subpart* function. Figure 5.12 shows the information displayed in a single execution of the *selected part name* function.

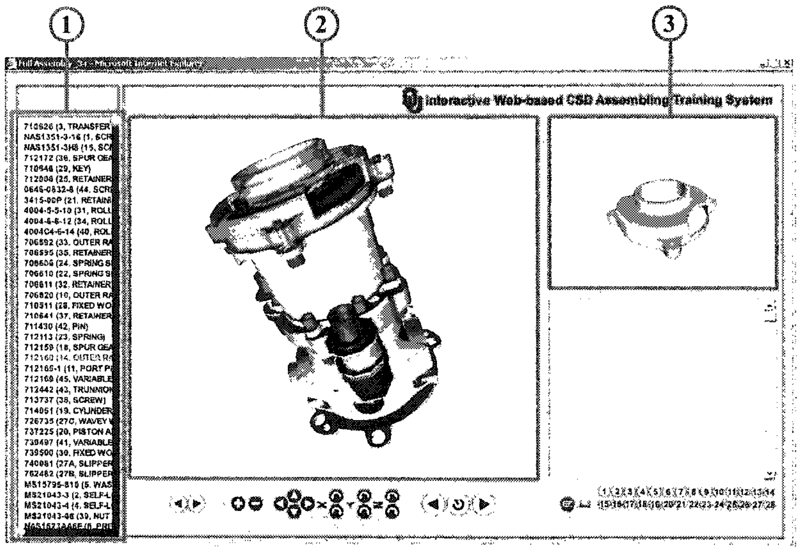


Figure 5.12: Information displayed in single *selected part name* function execution

Figure 5.13 shows the *selected part name* function mechanism flowchart. The mechanism of the *selected part name* function, which is associated only in the *Part Breakdown* section, is different compared to the *selected part assembly* function. As shown in the flowchart, the implementation of this mechanism begins when each particular part name in the *left menu area* is selected by users and the Lingo scripting language function associated with that particular part name calls and passes the part name as a parameter to other functions that perform consequence operations. Examples of these consequence operations include changing the color of the part name and part ID in the *left menu area*, making that particular part in the three-dimensional assembly model become transparent, or displaying an appropriate three-dimensional model in the *secondary window*. If no part name is selected then no function executes. Additionally, in the *Part Breakdown* section, there is no text instruction displays in the *text message window* because this section does not involve any assembly process step yet.

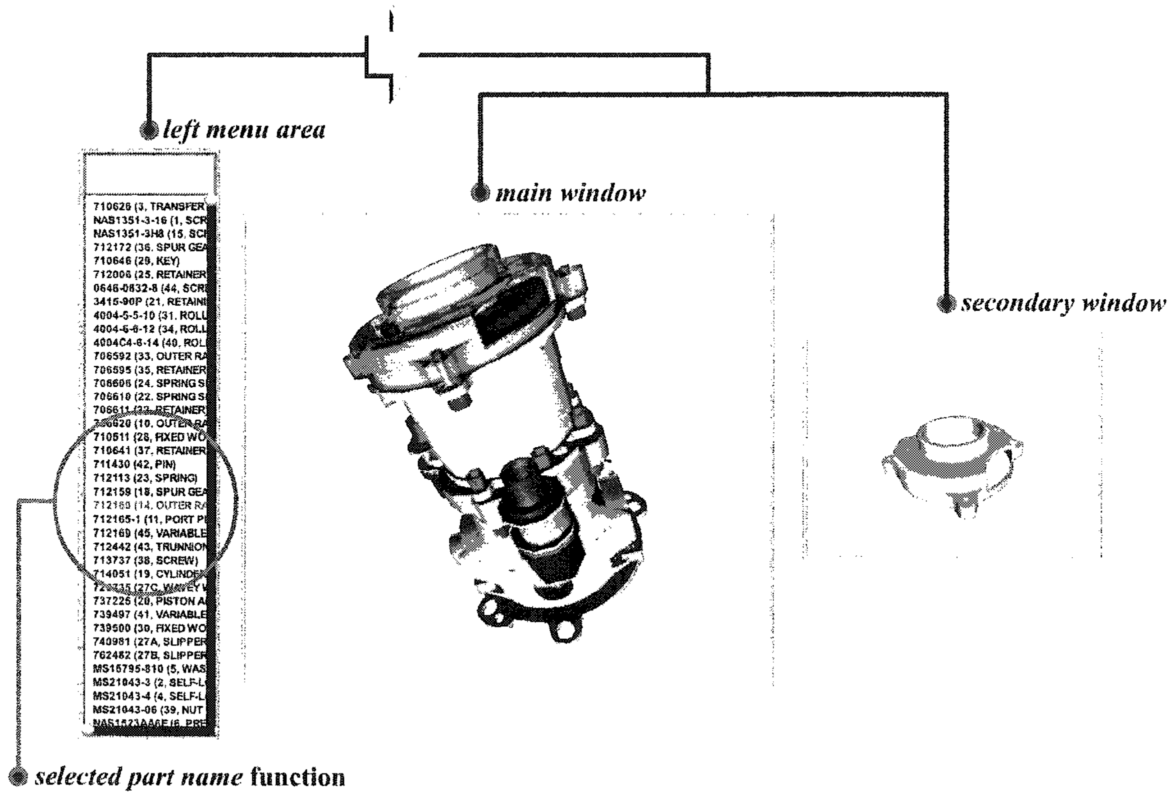
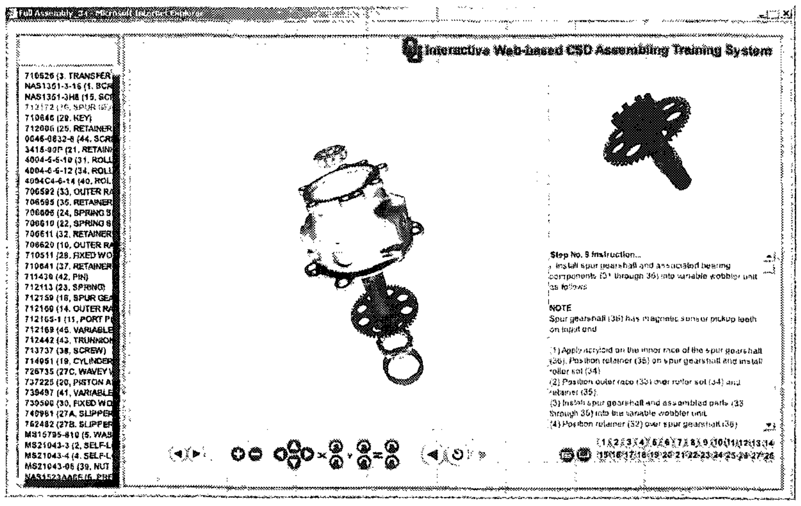


Figure 5.13: Flowchart of the selected part name function in *Part Breakdown*

2.1.2 Second Section: *Assembly Process Tutorial* section

After users have a solid understanding of the *LHPM Assembly* (illustrated part assembly) and all subparts contained in the *LHPM Assembly*, users advance to the next level of this training system to learn how to accurately assemble the *LHPM Assembly*. The second section, namely the *Assembly Process Tutorial* section, presents step-by-step assembly process of the *LHPM Assembly* through the use of animation. To closely imitate the real assembly process, three-dimensional models are employed for three-dimensional animation that can help users understand the concepts clearly.

The second section is made up of a number of learning frames or steps, with each consisting of an animation module that closely mimics the actual step-by-step assembly process being taught. This section also incorporates a low level of simulation that allows users to manipulate the tutorial, such as replaying an animation. This section allows users to study step-by-step assembly process using animation that users can replay anytime during that particular step or frame. An example screen of the *Assembly Process Tutorial* section is illustrated as shown in Figure 5.14.



Step No. 9 Instruction...
Install spur gearshaft and associated bearing components (31 through 36) into variable wobbler unit as follows

NOTE
Spur gearshaft (36) has magnetic sensor pickup teeth on input end

(1) Apply acryloid on the inner race of the spur gearshaft (36). Position retainer (35) on spur gearshaft and install roller set (34)
(2) Position outer race (33) over roller set (34) and retainer (35).
(3) Install spur gearshaft and assembled parts (33 through 36) into the variable wobbler unit.
(4) Position retainer (32) over spur gearshaft (36)

Figure 5.14: Example screen of the *Assembly Process Tutorial* section

This *Assembly Process Tutorial* section is followed up with an evaluation simulation, which is explained in detail later. In this section, each frame demonstrates how to assemble an assembly part step-by-step in sequence. Once a user begins this section, the corresponding assembly process animation will be displayed in the main working environment or the *main window*. The assembly process animation is sequentially displayed from the beginning to the end of the assembly process for each frame or step. The mechanism of displaying sequential step-by-step assembly process animations is implemented using Lingo scripting language. This is an important feature for this section in order to make certain that users are able to complete the assembly process step-by-step requirements as described in an official TO.

In addition, the text instruction description for each step appears in a scrollable *text message area* as well. This scroll bar is a built-in feature in Shockwave 3D (Director 3D). Users must learn the action required in a particular step by following the animation and instruction provided before moving on to the next step. To force users to learn the content in each frame before proceeding to the next frame, the forward button is inactive for a period of time to prevent users from moving forward without learning the content presented. The text instruction description that appears in the *text message area*, works collaboratively with the animated work environment in order to guide users' learning activities. It gives written explanations of the actions required of users. This allows users to understand more clearly how to complete the assembly process, rather than just blindly follow the animation step-by-step. It also benefits users in that the text material presented is identical to that presented in the official TO. There are fourteen steps in this *Assembly Process Tutorial* section. Figure 5.15 illustrates sample animation snapshots from the beginning to the end.

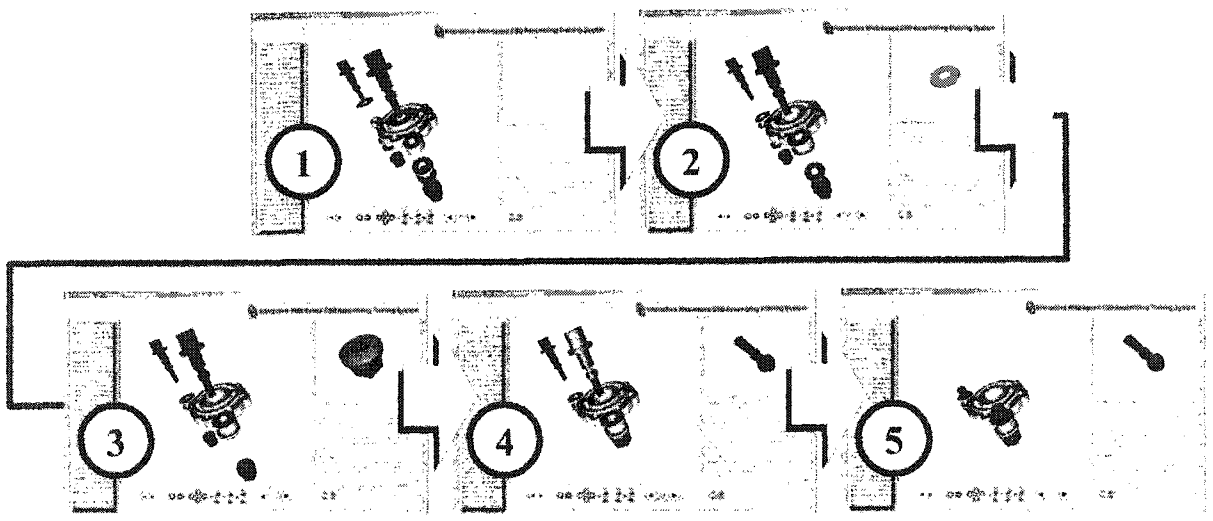
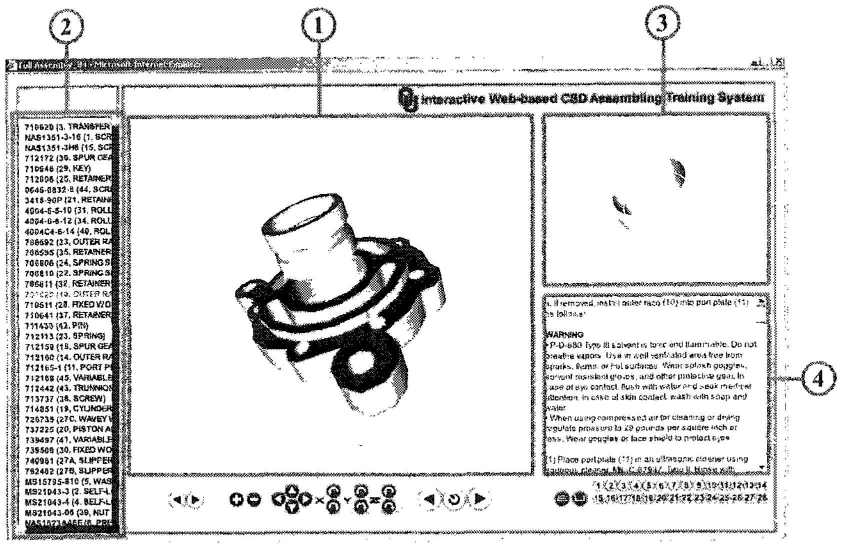


Figure 5.15: Samples of an animation process from the beginning to the end

Moreover, in the *Assembly Process Tutorial* section, the *secondary window* and the *left menu area* also provide additional information for each particular step of the assembly process animation displayed in the *main window*. The *secondary window* displays each particular subpart involved in that particular step of assembly process. This subpart is identical to the part that is in motion in the animation in the *main window*. Simultaneously, the *left menu area* highlights that particular part name and part ID as well. The part name and part ID of the part in motion changes its color from black to red to indicate that it is the part involved in the assembly process animation during this period of time. However, the *left menu area* itself is inactive, meaning it is not clickable. Figure 5.16 shows a sample of the information displayed in a single step of the assembly process animation.



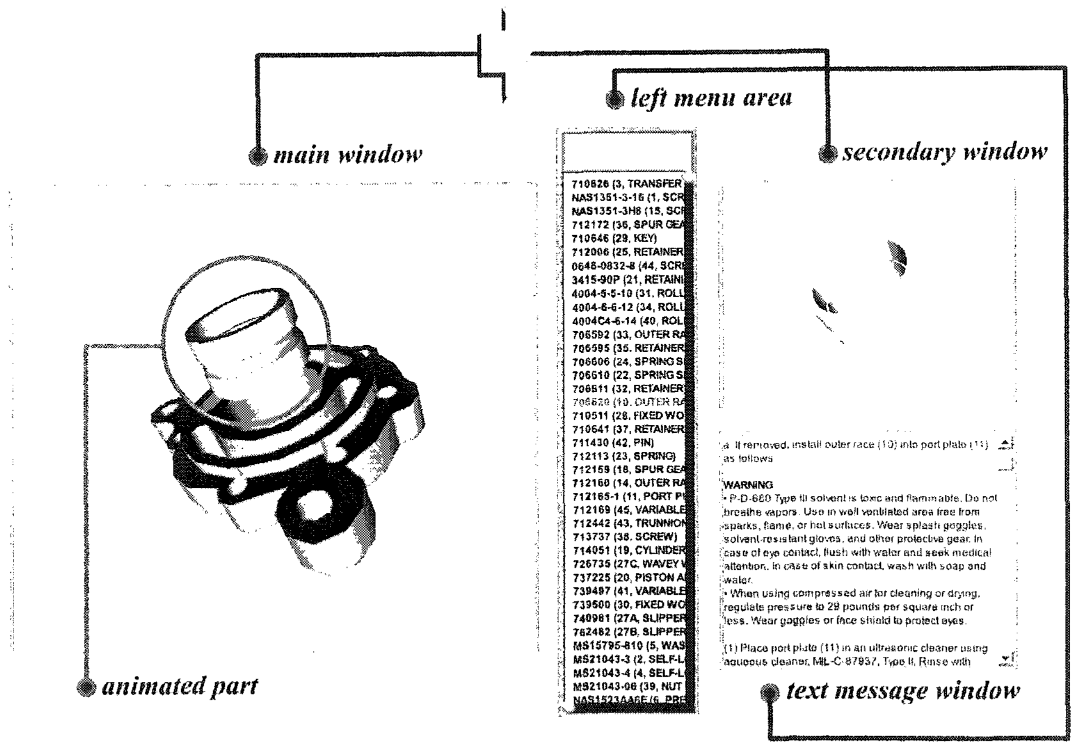
a. If removed, install outer race (10) into port plate (11) as follows:

WARNING
• P-D-680 Type III solvent is toxic and flammable. Do not breathe vapors. Use in well ventilated area free from sparks, flame, or hot surfaces. Wear splash goggles, solvent resistant gloves, and other protective gear. In case of eye contact, flush with water and seek medical attention. In case of skin contact, wash with soap and water.
• When using compressed air for cleaning or drying regulate pressure to 29 pounds per square inch or less. Wear goggles or face shield to protect eyes.

(1) Place port plate (11) in an ultrasonic cleaner using aqueous cleaner MIL-C-87937, Type II. Rinse with

Figure 5.16: Sample of information displayed in a single step of the animation

Also, a flowchart, as illustrated in Figure 5.17, shows the consequences of what happens when the function that plays the animation process is executed. As illustrated in the flowchart, the operations in the *Assembly Process Tutorial* section start with the motion of the part involved in a particular assembly step when users enter the frame and the *secondary window* shows that subpart individually. At the same time, the *text message window* displays text instruction description and the *left menu area* indicates the part name and part ID using a different color from the rest. In this section, users cannot manipulate three-dimensional assembly models through an input device because the action that appears in the *main window* is controlled by the function that plays the assembly animation. So, in order to avoid action overlap from users' control and function control, selected part function is inactive. However, users can still control the three-dimensional models that appear in the *secondary window* the same way they do in the *Part Breakdown* section.



```
710826 (3, TRANSFER
NAS1351-3-16 (1, SCR
NAS1351-3H8 (15, SCR
712172 (36, SPUR GEA
710646 (29, KEY)
712006 (25, RETAINER
0646-0832-8 (44, SCR
3415-90P (21, RETAINI
4004-5-5-10 (31, ROLL
4004-6-6-12 (34, ROLL
4004C4-6-14 (40, ROL
706592 (33, OUTER RA
706595 (35, RETAINER
706606 (24, SPRING S
706610 (22, SPRING S
706611 (32, RETAINER
706620 (10, OUTER RA
710511 (28, FIXED WO
710641 (37, RETAINER
711430 (42, PIN)
712113 (23, SPRING)
712159 (18, SPUR GEA
712160 (14, OUTER RA
712165-1 (11, PORT P
712169 (45, VARIABLE
712442 (43, TRUNNION
713737 (38, SCREW)
714051 (19, CYLINDER
726735 (27C, WAVEY W
737225 (20, PISTON A
739497 (41, VARIABLE
739500 (30, FIXED WO
740961 (27A, SLIPPER
762482 (27B, SLIPPER
MS15795-810 (5, WAS
MS21043-3 (2, SELF-L
MS21043-4 (4, SELF-L
MS21043-06 (39, NUT
NAS1523AA6E (6, PRE
```

a. If removed, install outer race (10) into port plate (11) as follows

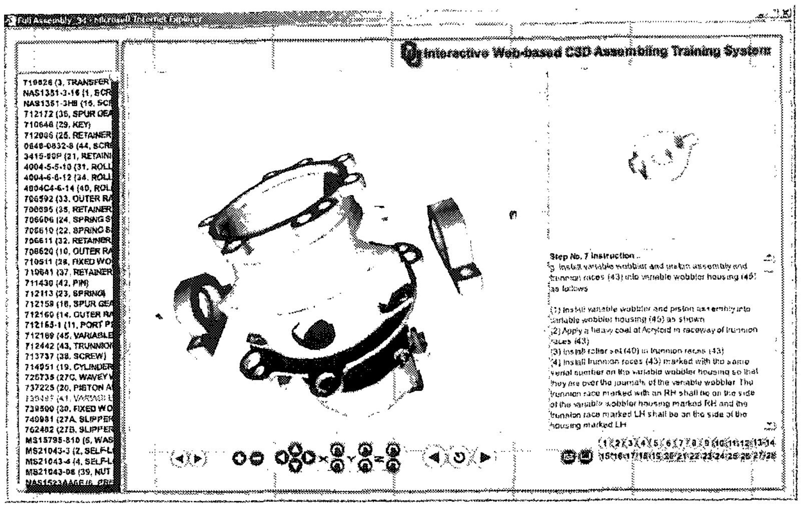
WARNING
• P-D-680 Type III solvent is toxic and flammable. Do not breathe vapors. Use in well ventilated area free from sparks, flame, or hot surfaces. Wear splash goggles, solvent-resistant gloves, and other protective gear. In case of eye contact, flush with water and seek medical attention. In case of skin contact, wash with soap and water.
• When using compressed air for cleaning or drying, regulate pressure to 29 pounds per square inch or less. Wear goggles or face shield to protect eyes.

(1) Place port plate (11) in an ultrasonic cleaner using aqueous cleaner, MIL-C-87937, Type II, Rinse with

Figure 5.17: Flowchart shows the consequences of what happens in the second section

2.1.3 Third Section: *Assembly Process Evaluation* section

After completing the first two sections, users are expected to understand how to correctly perform the actual assembly process. However, before performing any real assembly process activities, users should first be evaluated using assembly process simulation. The third section is designed to perform step-by-step assembly process. Figure 5.18 is illustrates an example screen of the *Assembly Process Evaluation* section.



Step No. 7 Instruction ..
g. Install variable wobbler and piston assembly and trunnion races (43) into variable wobbler housing (45) as follows

(1) Install variable wobbler and piston assembly into variable wobbler housing (45) as shown
(2) Apply a heavy coat of Acryloid in raceway of trunnion races (43)
(3) Install roller set (40) in trunnion races (43)
(4) Install trunnion races (43) marked with the same serial number on the variable wobbler housing so that they are over the journals of the variable wobbler. The trunnion race marked with an RH shall be on the side of the variable wobbler housing marked RH and the trunnion race marked LH shall be on the side of the housing marked LH

Figure 5.18: Example of a screen in the *Assembly Process Evaluation* section

When users finish learning the right way to perform assembly process, the next step is to assess the knowledge users obtained from the previous section. This step is necessary to ensure that users are ready to correctly perform assembly process as expected before users begin their training. However, to evaluate users using the actual part assembly is costly and risky. Therefore, the *Assembly Process Evaluation* section is introduced as an evaluation tool for this training using simulations. This section evaluates users' performance in assembly process step-by-step, which is similar to the section in which users learn the correct assembly process. Three-dimensional models are employed for three-dimensional simulations in order to make users feel as though they were performing the assembly process using actual part assembly. Three-dimensional simulation can help users during the assessment because users can perform, view, and understand the assembly process using a three-dimensional view.

This *Assembly Process Evaluation* section is incorporated with a variety of high-level interactivity simulations. This simulation module's design is based on the animation module introduced in the *Assembly Process Tutorial* section and is used to track users' actions. Users have an opportunity to perform an assembly process step-by-step as presented in the previous section. Users have to correctly move a set of models to their location in the assembly part according to a particular step of an assembly. Users have to sequentially perform the assembly procedure, meaning that if there is more than a single model involved in a particular step, users have to correctly finish the model simulation in order to move on to the next model and to appropriately complete the assembly process. Activating simulated parts in sequence is controlled by Lingo scripting language.

Whenever a wrong movement is made due to a users' misunderstanding or by accident, that particular model will bounce back to its original position to let users know that the location is not the correct location. Users must then try again until the correct location is discovered. Users will not be able to proceed to the next movable model in that step unless the model in the preceding order has been placed in the correct location. This immediate response feature alerts users to errors and guides them to take the right action, thereby helping users realize and understand the mistake better. These model functions operate with the use of Lingo script language as well. The idea behind these functions is that when the model is placed, the first function calculates the distance between that particular position and the correct position, and then determines if the difference is considered acceptable in the maximum distance allowed by the function. If it is acceptable, the second function will place that part in the correct position, make that part inactive, and then activate the next part in the sequence. If it is not acceptable, the second function will move that part back to its original position to indicate that users have made a wrong move and must try again. During each period of time in each step of the assembly process, only one part is active or movable. Figure 5.19 illustrates samples of models in a simulation process when the position is correct and incorrect.

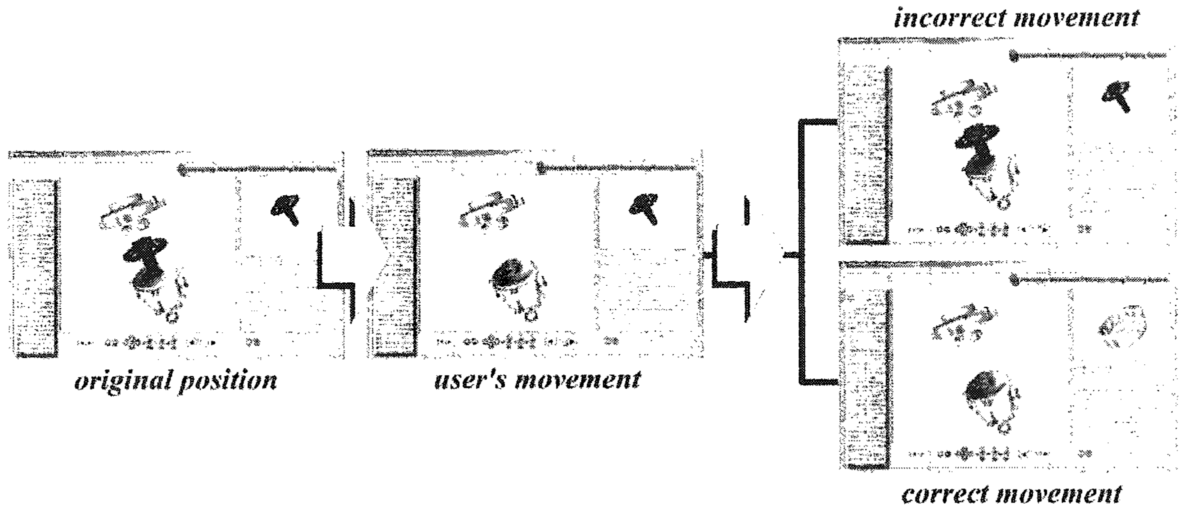


Figure 5.19: Samples of a simulation process in both correct and incorrect situations

As mentioned, there are four possibilities in this section; (1) users make the right move and then proceed to the next model where multiple models are involved in a single step, (2) users make the right move to finish the step and then proceed to the next step where a single model is involved in a single step, (3) users make the wrong move and then have to stay at the same model for another try where multiple models are involved in a single step, and (4) users make the wrong move and then have to stay in the same step for another try with the same model where a single model is involved in a single step. Figure 5.19 (1 and 3 possibilities) and 5.20 (2 and 4 possibilities) illustrate examples of steps to illustrate the four possibilities.

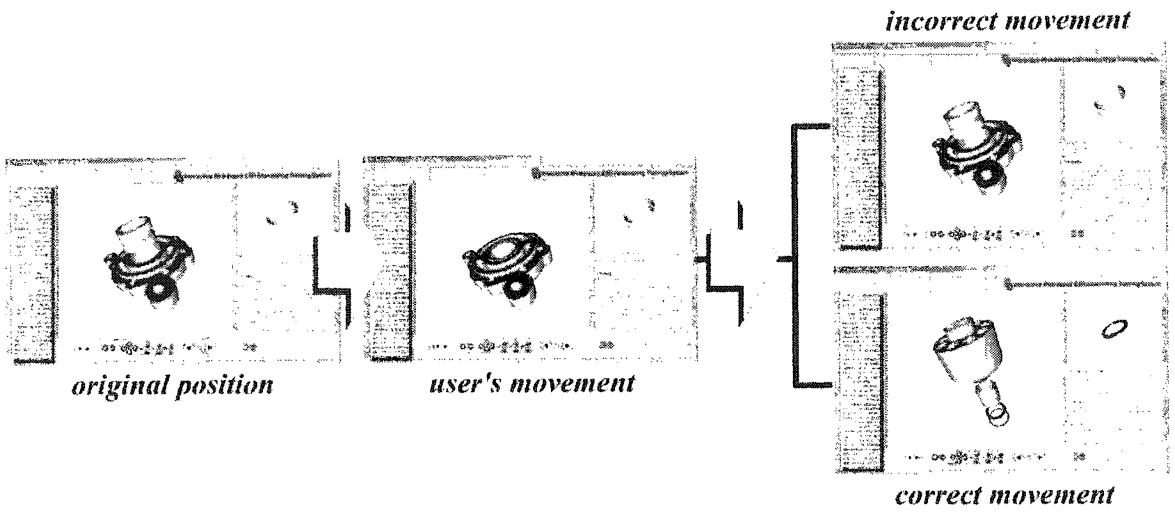


Figure 5.20: Sample of all possibilities that can happen in *Assembly Process Evaluation*

The *Assembly Process Evaluation* section appears after reaching the end of the second section in order to give users a chance to assess their own understanding of an entire assembly process. Therefore, if users feel uncertain about being able to perform the right procedures while in the third section, they have the option to go back to the previous sections anytime to review either the tutorial section from the beginning or a particular step in performing the assembly process. Figure 5.21 illustrates sample simulation snapshots from the beginning to the end.

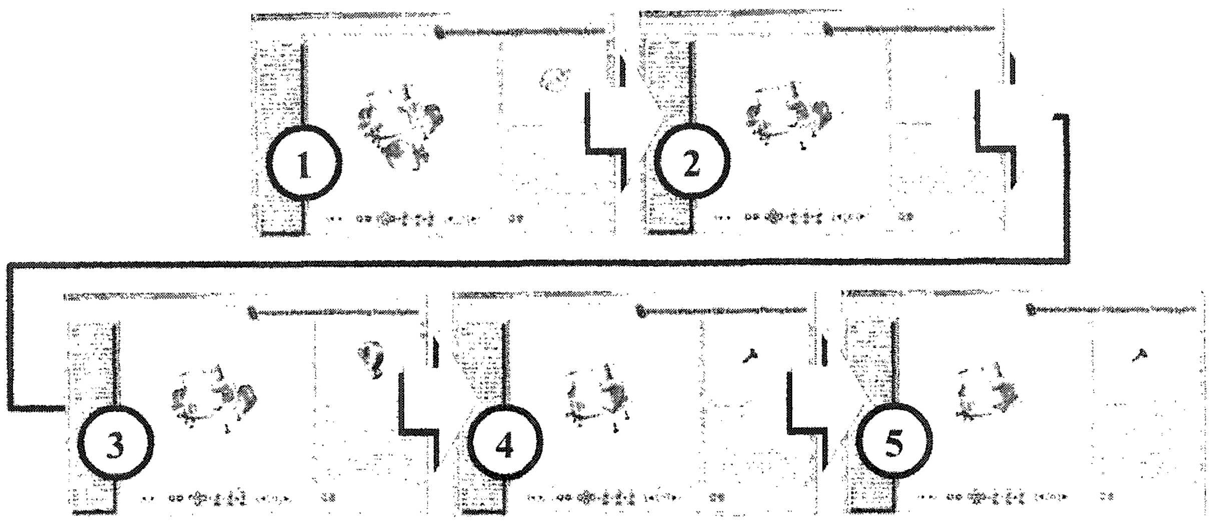
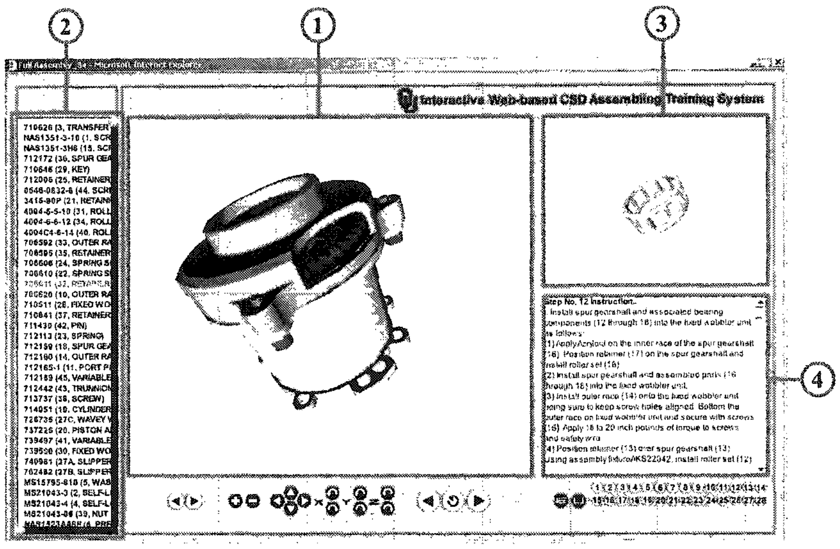


Figure 5.21: Sample of a complete single step simulation process

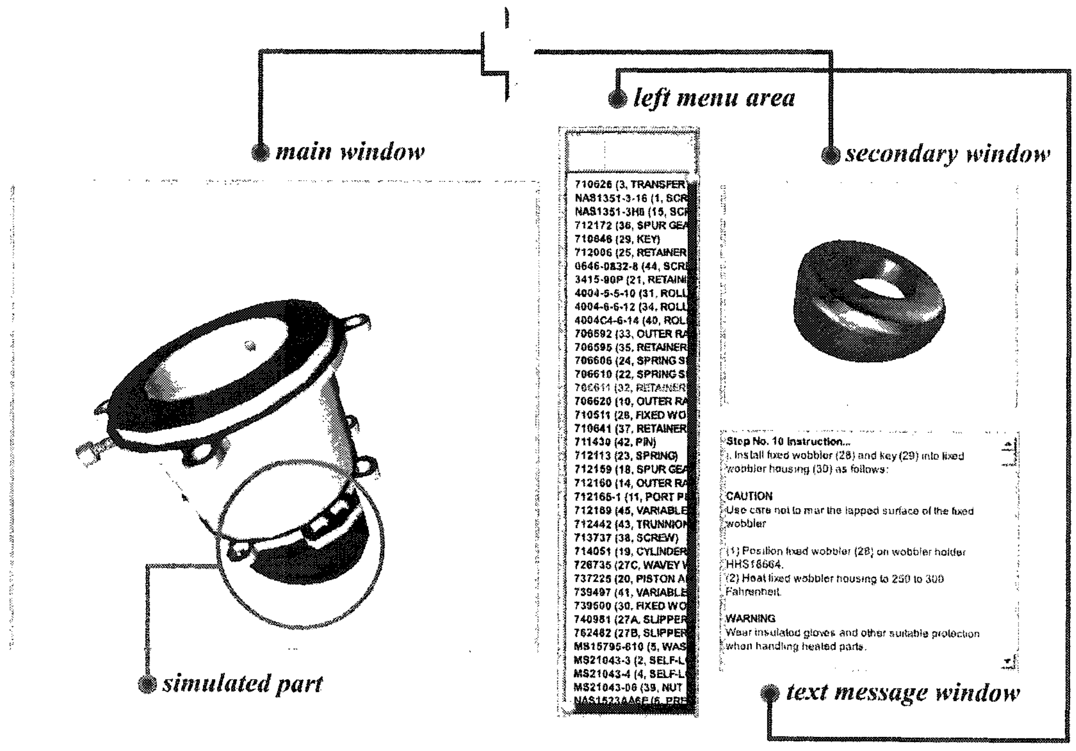
Additionally, like the mechanism of the *Assembly Process Tutorial* section, the *secondary window*, *left menu area*, and *text message window* present supplementary information about the simulated part that is active in that particular assembly simulation process, exactly the same way as presented in the second section as well. The *secondary window* displays a specific part that refers to a particular simulation step, while text instruction description is displayed in the *text message window* and part name is highlighted in the *left menu area*. However, the *left menu area* itself is inactive. Figure 5.22 demonstrates a sample of the information displayed in a single step of the assembly process simulation. The flowchart shows the consequences of what happens when the function that starts the simulation process is executed, as shown in Figure 5.23.



710626 (3, TRANSFER
NAS1351-3-16 (1, SCR
NAS1351-3H8 (15, SCR
712172 (36, SPUR GEA
710646 (29, KEY)
712006 (25, RETAINER
0646-0832-8 (44, SCR
3415-90P (21, RETAIN
4004-5-5-10 (31, ROLL
4004-6-6-12 (34, ROLL
4004C4-6-14 (40, ROL
706592 (33, OUTER RA
706595 (35, RETAINER
706606 (24, SPRING S
706610 (22, SPRING S
706611 (32, RETAINER
706620 (10, OUTER RA
710511 (28, FIXED WO
710641 (37, RETAINER
711430 (42, PIN)
712113 (23, SPRING)
712159 (18, SPUR GEA
712160 (14, OUTER RA
712165-1 (11, PORT P
712189 (45, VARIABLE
712442 (43, TRUNNION
713737 (38, SCREW)
714051 (19, CYLINDER
726735 (27C, WAVEY W
737225 (20, PISTON A
739497 (41, VARIABLE
739500 (30, FIXED WO
740981 (27A, SLIPPER
762482 (27B, SLIPPER
MS15795-818 (5, WAS
MS21043-3 (2, SELF-L
MS21043-4 (4, SELF-L
MS21043-06 (39, NUT
NAS1523AA6E (6, PRE

Step No. 12 Instruction..
1. Install spur gearshaft and associated bearing components (12 through 18) into the fixed wobbler unit as follows:
(1) Apply Acryloid on the inner race of the spur gearshaft (16). Position retainer (17) on the spur gearshaft and install roller set (18)
(2) Install spur gearshaft and assembled parts (16 through 18) into the fixed wobbler unit.
(3) Install outer race (14) onto the fixed wobbler unit being sure to keep screw holes aligned. Bottom the outer race on fixed wobbler unit and secure with screws (15). Apply 16 to 20 inch pounds of torque to screws and safety wire.
(4) Position retainer (13) over spur gearshaft (13). Using assembly fixture/KS22042, install roller set (12)

Figure 5.22: Sample of the information displayed in a single step of the simulation



710626 (3, TRANSFER
NAS1351-3-16 (1, SCR
NAS1351-3H8 (15, SCR
712172 (36, SPUR GEA
710646 (29, KEY)
712006 (25, RETAINER
0646-0832-8 (44, SCR
3415-90P (21, RETAIN
4004-5-5-10 (31, ROLL
4004-6-6-12 (34, ROLL
4004C4-6-14 (40, ROL
706592 (33, OUTER RA
706595 (35, RETAINER
706606 (24, SPRING S
706610 (22, SPRING S
706611 (32, RETAINER
706620 (10, OUTER RA
710511 (28, FIXED WO
710641 (37, RETAINER
711430 (42, PIN)
712113 (23, SPRING)
712159 (18, SPUR GEA
712160 (14, OUTER RA
712165-1 (11, PORT P
712189 (45, VARIABLE
712442 (43, TRUNNION
713737 (38, SCREW)
714051 (19, CYLINDER
726735 (27C, WAVEY W
737225 (20, PISTON A
739497 (41, VARIABLE
739500 (30, FIXED WO
740981 (27A, SLIPPER
762482 (27B, SLIPPER
MS15795-810 (5, WAS
MS21043-3 (2, SELF-L
MS21043-4 (4, SELF-L
MS21043-06 (39, NUT
NAS1523AA6E (6, PRE

Step No. 10 Instruction...
1. Install fixed wobbler (28) and key (29) into fixed wobbler housing (30) as follows:

CAUTION
Use care not to mar the lapped surface of the fixed wobbler

(1) Position fixed wobbler (28) on wobbler holder HHS18664.
(2) Heat fixed wobbler housing to 250 to 300 Fahrenheit.

WARNING
Wear insulated gloves and other suitable protection when handling heated parts.

Figure 5.23: Flowchart shows the consequences of what happens in the third section

2.2 Navigation System Design

The primary focus of this learning system is to educate and make users understand their responsibilities. To assist in these goals, it is important that the actual lesson navigation is simple and flexible so that the users' time is not wasted researching how to use the features. A graphical user interface with a push-button system is used so that the functionality of each button can be understood easily through visual identification. In order for easy browsing, two sets of navigation systems are utilized in this training system; they are system navigation and model navigation. The navigation system is implemented using Lingo scripting language.

The placement of a navigation controls area can facilitate users in easy navigation throughout an entire system's presentation. The navigation is located at the bottom of each lesson screen so that the buttons do not interfere with the presentation of material. An example of a lesson frame that illustrates only the navigation system area or the area of user control features is shown in Figure 5.24.

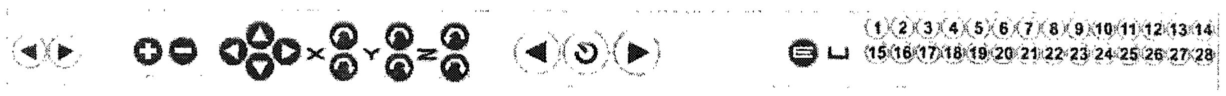


Figure 5.24: Area of user control features

To focus the users' attention on the learning material, a simple navigation system that can provide the required level of control and can keep the presentation in a linear process with bi-directional navigation is implemented. These navigation controls are designed to accommodate the nonlinear style of learning as well as the linear style of learning. The level of control must also provide the option for users to review present content or any topic while in a given location in the system. When users are free to navigate around the learning material for different contents and the same document in more than one way, users have the advantage of being able to acquire concepts that are highly related to one another. This feature also benefits users who prefer to selectively learn certain content in the environment and is incorporated to force users to learn the material in a linear process, which is required to properly finish the assembly part. However, the environment also provides users with a nonlinear learning process in order to provide an opportunity for users to review the material already learned. It allows users to revisit a particular frame in order to review the material contained in that frame.

Three main functional navigation buttons are provided for users to control the operation of lessons. These functional navigation buttons are forward, repeat and backward. These buttons are arranged in a chronological order based on their functionality, so that the stereotyped human behavior to click the right most button to go next and to click the left most button to go back is supported. Figure 5.25 shows only the main navigation buttons.

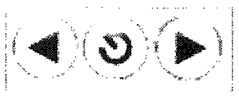


Figure 5.25: Main navigation buttons

The forward button is used to proceed to the next frame or step. This button is activated after a user is in the frame for five seconds so that the user must view the contents inside each object at least once before proceeding to the next concept. This mechanism restricts users from quickly moving through the lesson without viewing the contents, and is implemented so that users are forced to go through the material in a linear process. All steps must be done in order, but completed steps can be reviewed in any order. The backward button provides users with the ability to return to the previous frame or step. The repeat button is located in the middle of the forward and backward buttons because its functionality lies between the other two features. The repeat button is incorporated to provide users with the option to review the material inside a particular frame again before proceeding. Additional navigation buttons are shown in Figure 5.26.

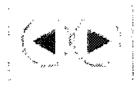


Figure 5.26: Additional navigation buttons

The system also includes an additional set of small forward and backward buttons apart from the standard buttons available for users. These buttons are additional features provided to support users' moves forward or backward at anytime. When a user does not intend to complete a lesson frame by frame, they are provided with this feature to quickly browse through the contents. This is a feature that appears as an option even though users should completely view all the frames. These buttons are separated from the primary navigation system and are located in the left corner of the navigation bar. They are not integrated with the regular buttons because of their difference in functionality. The frame buttons provide users with more convenience and information for each particular frame. The number on the buttons shows users exactly how many frames are included in a particular environment. For this particular subassembly (the *LHPM Assembly*), there are a total of thirty frames, including two frames in the first section that represent E (Exploded View) and U (Unexploded View) buttons and fourteen frames in each of the second and third sections. A set of frame buttons is illustrated in Figure 5.27.

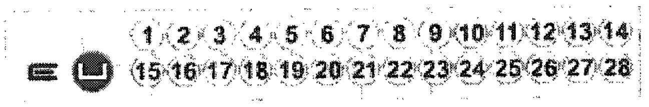


Figure 5.27: Frame navigation buttons

There is also a set of model control buttons that allow users to rotate, translate, and zoom in and out of the model to gain a better understanding of the part detail and assembly process. This set of buttons is located under the *main window* and is placed between the standard navigation and additional navigation buttons. The model control buttons are used directly by users to control the movements of the three-dimensional assembly model in the *main window* only. At anytime during the learning process users can manipulate the three-dimensional assembly model, such as rotating, moving, minimizing, or maximizing the model to obtain more information related to the model, assembly process animation, or simulation, as well as to better control the simulated model in the *Assembly Process Evaluation* section. The available movements are rotation and translation. Rotation is based on an x, y, and z coordinate system incorporated into the three-dimensional model itself. The combination of multiple axes rotation at the same time is not provided. There are six buttons associated with this rotation, which are divided into plus and minus rotation along x, y, and z-axes. These buttons are shown in Figure 5.28.

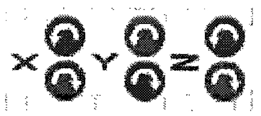


Figure 5.28: Rotation movement buttons

Translation movement is only in two dimensions, which are divided into up, down, left, and right translations that allow users to only move the three-dimensional assembly model located in the *main window* in four directions. Four buttons accompany the rotation movement feature, as illustrated in Figure 5.29. To zoom in and out of the three-dimensional assembly model (3D file) in the *main window*, two buttons are provided for zoom in and out separately. Figure 5.30 shows the zoom in and out buttons provided in the learning system.

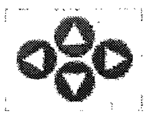


Figure 5.29: Translation movement buttons

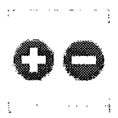


Figure 5.30: Zoom in and out buttons

As mentioned, these model control buttons are used to control the three-dimensional assembly model in the main window only. However, three-dimensional part models can be controlled by users as well, but only rotation movement is provided. Furthermore, to manipulate three-dimensional part models, no additional button is required. The rotation is controlled through an input device, such as a mouse and touch pad (laptop), so that users can see what the part looks like from all angles. Figure 5.31 shows sample screens of three-dimensional model movement in the *secondary window*. All of these buttons are implemented using Lingo scripting language.

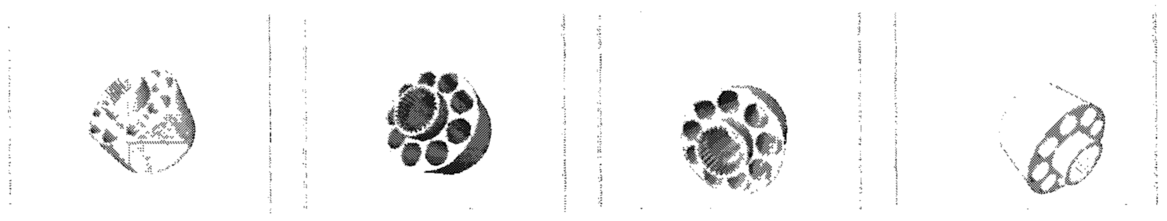


Figure 5.31: Sample screens of movement in the *secondary window*

3D Dynamics System

With many applications in the field of education, multimedia technologies are particularly well suited for implementation in engineering education. Many topics in engineering are abstract and difficult to visualize, such as work and energy. Others are difficult to visualize due to the nature of the dynamics involved, such as the *Kinematics of Rigid Bodies*. Concepts are usually taught by developing abstract mathematical models and fundamental physical principles, which are then employed to solve practical problems. Connecting abstract models with practical situations and presenting time as an independent variable are difficult aspects of the engineering curriculum, especially at the introductory level. With its ability to integrate motion into the presentation, multimedia is ideal for addressing difficult topics in *Engineering Dynamics*, such as *Kinetics of Particles* (force, acceleration, work, energy, impulse, and momentum), the *Kinematics of Rigid Bodies in Two and Three Dimensions*, and *Vibrations*.

Although a large percentage of *Dynamics* problems in engineering can be solved by the principles of plane motion (two dimensions), modern developments have focused increasing attention on problems that call for the analysis of motion in three dimensions. Inclusion of a third dimension adds considerable complexity to the *Kinematics* and *Kinetics* relationships. Not only does the added dimension introduce a third component to vectors that represent force, linear velocity, linear acceleration, and linear momentum, but the introduction of the third dimension adds the possibility of two additional components of vectors representing angular quantities, including moments of forces, angular velocity, angular acceleration, and angular momentum. It is in three-dimensional motion that the full power of vector analysis is utilized.

A good background in the *Dynamics* of plane motion (*Two-dimensional Dynamics*) is extremely useful in the study of *Three-dimensional Dynamics*, where the approach to problems and many of the terms are the same as or analogous to those in two dimensions. If the study of *Three-dimensional Dynamics* is undertaken without the benefit of prior study of plane-motion dynamics (*Two-dimensional Dynamics*), more time will be required to master the principles and to become familiar with the approach to problems. Indeed, the possibility of working on *Three-dimensional Dynamics* effectively without knowing *Two-dimensional Dynamics* is unlikely.

The materials presented in this training system is not intended as a complete development education of three-dimensional motion of rigid bodies (*Three-dimensional Rigid Body Motion*), but merely as a basic introduction to the subject. This introduction should, however, be sufficient to solve many of the more common problems in three-dimensional motion and also to lay the foundation for more advanced study. Proceeding within this subject is the same way as proceeding within particle motion and rigid-body motion (*Two-dimensional Particle and Rigid Motion*), by first examining the necessary *Kinematics* and then continuing to the *Kinetics*.

The 3D Dynamics System is intended to serve as a supplement to textbooks and lectures for students taking an undergraduate *Engineering Dynamics* course in the topic of *Three-dimensional Rigid Body Motion*. It is, however, in no way intended to replace the textbook or lectures. Moreover, at the University of Oklahoma, a web-based course system on the topic of *Engineering Dynamics* (www.eCourses.ou.edu) is another main resource to assist students in learning this topic more effectively. Therefore, the 3D Dynamics System can be considered an additional web-based tool for this web-based course system as well. The 3D Dynamics System is mainly designed for the distribution of information interactively about a specific topic in *Engineering Dynamics* on the web. The system is self-contained and fully integrated so that students have full control over navigating through the content and exploring the simulations. This full control capability, along with the use of animations and other media provided, can improve students' understanding of the concepts presented. *Three-dimensional Rigid Body Motion* is tremendously suitable for the demonstration of this research because three-dimensional visualization is essential to assist in explaining the concept clearly.

The system, which is considered an extended module or topic of the EM/C135 System, is composed of two main lessons illustrating specific concepts in *Three-dimensional Rigid Body Motion*, including *Three-dimensional Kinematics of a Rigid Body* and *Three-dimensional Kinetics of a Rigid Body*. Each lesson consists of a number of objectives in order to cover all the major topics in a fashion that is similar to that of a typical *Engineering Dynamics* textbook. The Three-*dimensional Kinematics of a Rigid Body* lesson consists of three topics or objectives; they are *Motion About a Fixed Point*, *Relative-Motion Analysis Using Translating Axes*, and *Relative-Motion Analysis Using Translating and Rotating Axes*. The *Three-dimensional Kinetics of a Rigid Body* lesson presents two topics, including *Moments and Products of Inertia* and *Equations of Motion*. Figure 5.32 illustrates the Structure of topics covered in the 3D Dynamics System.

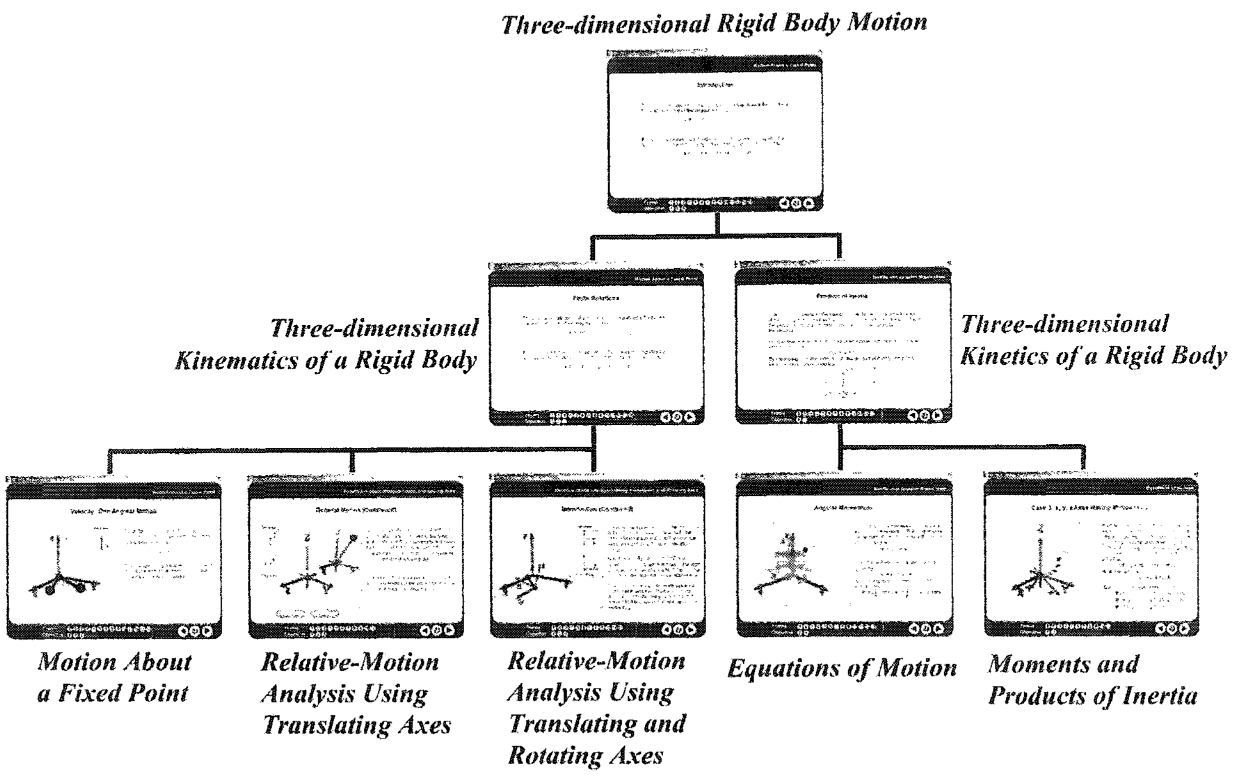

Three-dimensional Rigid Body Motion
Three-dimensional Kinematics of a Rigid Body
Three-dimensional Kinetics of a Rigid Body
Motion About a Fixed Point
Relative-Motion Analysis Using Translating Axes
Relative-Motion Analysis Using Translating and Rotating Axes
Equations of Motion
Moments and Products of Inertia

Figure 5.32: Structure of topics covered in the 3D Dynamics System

The system covers all the major subtopics of the area of *Three-dimensional Rigid Body Motion* presented in a typical *Engineering Dynamics* textbook, although it does not cover every topic in every textbook. However, the subtopics presented in this system are included in most *Engineering Dynamics* textbooks available as learning resources. The concepts and examples addressed in this system are based on the same topics and details of the ninth edition *Engineering Mechanics: Dynamics*, by R. C. Hibbler. This textbook is one of the most popular and well-known textbooks used in teaching *Engineering Dynamics* in many colleges all around the world. The basic unit used in this system (in US unit not SI unit) is the same as the basic unit used in Hibbler's textbook. Furthermore, this textbook is widely used by professors teaching this subject at the University of Oklahoma as well.

Each topic is designed to illustrate one particular concept, and each concept is split into three areas, including *Theory*, *Simulation Activity*, and *Problem*. These three areas are organized so as to provide students with a linear learning process in order to help students clearly understand the concept. Additionally, each topic is composed of various frames to cover required material contents. All theory required to solve the problem at the end of each topic is presented. The scope of concepts staged in this system is considered a basic level of knowledge that students should have learned in order to explore the topic of *Engineering Dynamics* at an advanced level. The first and third areas make extensive use of animations, sound, graphics, text, and equations to present the material. The first area introduces a specific concept to the student, while the second area is a fully interactive simulation of the problem presented in the third area. At anytime a student can invoke the simulation for an experiential representation of the problem. It should be noted that students are required to answer questions before moving to the next concept. This system is intended to serve as a tutorial where the progress of the student is monitored or controlled. Figure 5.33 illustrates the linear learning structure of the 3D Dynamics System.

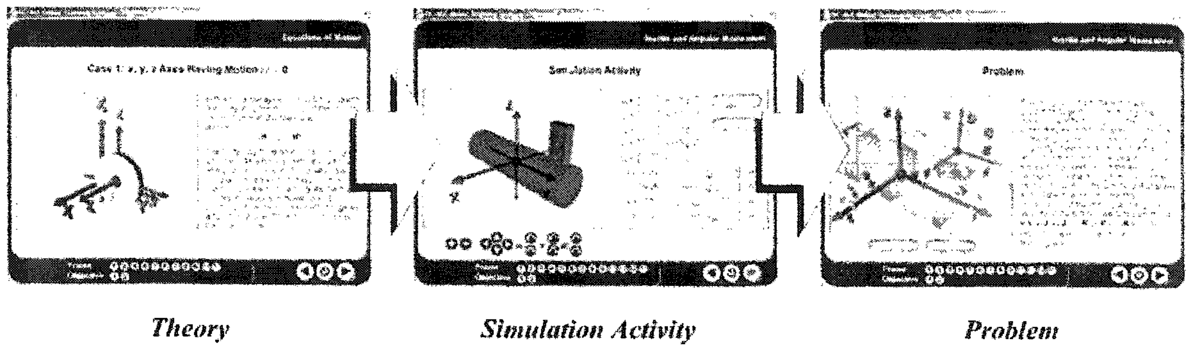
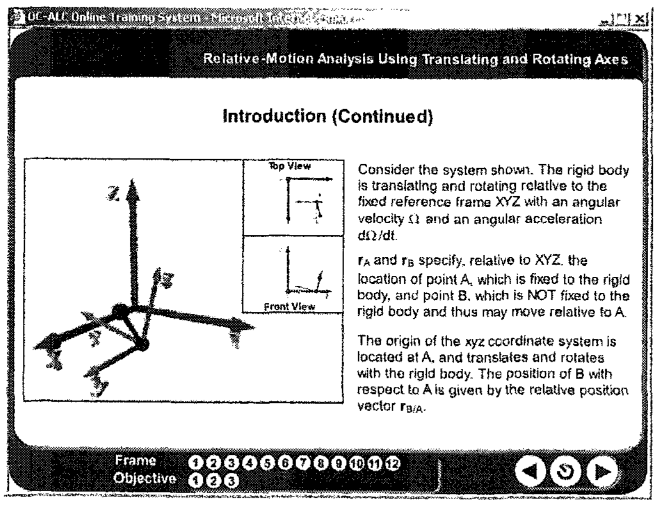


Figure 5.33: Linear structure of the three areas for each topic (objective)

The learning system's organization and lesson operation are the same as the EM/C135 System discussed in the previous chapter. The 3D Dynamics System has the same features available in the EM/C135 System, such as collecting evaluation scores, displaying a congratulations page at the end of each lesson, etc. However, the learning content layout is different because the information presented in the 3D Dynamics System is based on heavily used equations and three-dimensional graphics, diagrams, animations, and simulations. First, the location of media, such as graphics, diagrams, animations, and simulations, is consistently on the left side in order to have enough room on the other side of the screen to fit together the text description information, equation, and equation description information. This is important in order for students to understand the theory that explains the concept correctly and completely without any information breaking down. Figure 5.34 illustrates a sample screen of the 3D Dynamics System with media (animation, graphics, or diagram) on the left side.


UC-ALC Online Training System - Microsoft Internet Explorer
Relative-Motion Analysis Using Translating and Rotating Axes
Top View
Front View
Z
Frame ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫
Objective ① ② ③

Introduction (Continued)

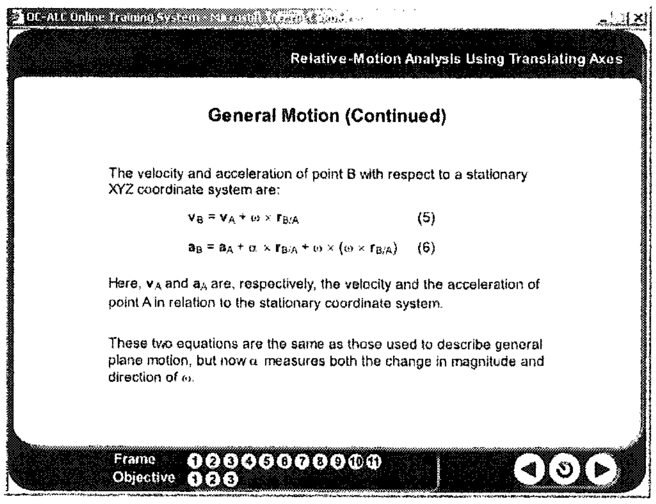
Consider the system shown. The rigid body is translating and rotating relative to the fixed reference frame XYZ with an angular velocity Ω and an angular acceleration $d\Omega/dt$.

r_A and r_B specify, relative to XYZ, the location of point A, which is fixed to the rigid body, and point B, which is NOT fixed to the rigid body and thus may move relative to A.

The origin of the xyz coordinate system is located at A, and translates and rotates with the rigid body. The position of B with respect to A is given by the relative position vector $r_{B/A}$.

Figure 5.34: Sample screen of the 3D Dynamics System with media

Another issue is the size of media that covers about half of the presentation material in the learning content layout, whenever it is incorporated. The main reason behind this is the significance of media that assists in explaining the concepts as well as text and equation information. The appearance of media in this learning system is in three dimensions that can help students gain the knowledge more effectively as one of the main objectives of this research. The animations used in the *Theory* and *Problem* areas keep playing as a loop until students leave the frames. The diagrams presented in the *Theory* area rotate for students to view all possible angles. However, media is not the only important component. Text and equation information are essential elements as well. Figure 5.35 shows a sample screen of the 3D Dynamics system with only text and equation information.



Relative-Motion Analysis Using Translating Axes

General Motion (Continued)

The velocity and acceleration of point B with respect to a stationary XYZ coordinate system are:

$$\mathbf{v}_B = \mathbf{v}_A + \omega \times \mathbf{r}_{B/A} \quad (5)$$

$$\mathbf{a}_B = \mathbf{a}_A + \alpha \times \mathbf{r}_{B/A} + \omega \times (\omega \times \mathbf{r}_{B/A}) \quad (6)$$

Here, $\mathbf{v}_A$ and $\mathbf{a}_A$ are, respectively, the velocity and the acceleration of point A in relation to the stationary coordinate system.

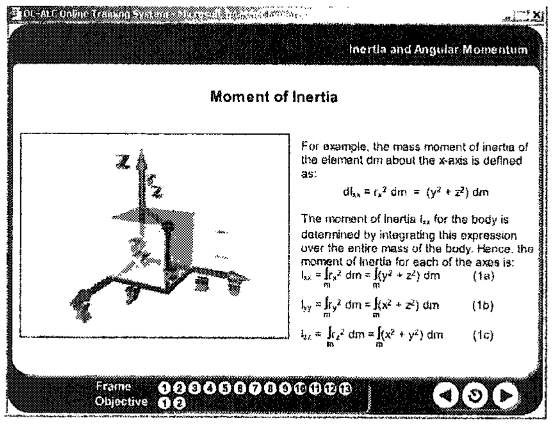
These two equations are the same as those used to describe general plane motion, but now α measures both the change in magnitude and direction of ω.

Figure 5.35: Sample screen of the 3D Dynamics System without media

1. Theory

In the *Theory* area, a specific concept in *Three-dimensional Rigid Body Motion* is presented in sufficient detail to understand and be able to solve the problem. This area generally ranges from four to eight frames and makes extensive use of graphics, diagram, animation, sound, and text to help provide an intuitive understanding of difficult concepts. In the *Theory* area, the scope of the theory is complete; however, the presentation is not as detailed as that found in a textbook. Emphasis is placed on presenting the theory in a concise and easy-to-use form so that students can quickly grasp its application to the problem.

All graphics, diagrams, and animations included in this area are represented in three-dimension. These media types are developed using primarily three-dimensional applications, such as *Swift3D* and *3ds Max*, along with other multimedia technologies such as *Flash* and *Photoshop*. Most graphics and animations are shown in a perspective view that incorporates the top, front, and side views in order to explain the concepts successfully. Samples of graphics, diagram, and animations are illustrated in Figures 5.36a, 5.36b, 5.37a, and 5.7b.


OC-ALC Online Training System
Inertia and Angular Momentum
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13
Objective 1 2

Moment of Inertia

For example, the mass moment of inertia of the element dm about the x-axis is defined as:

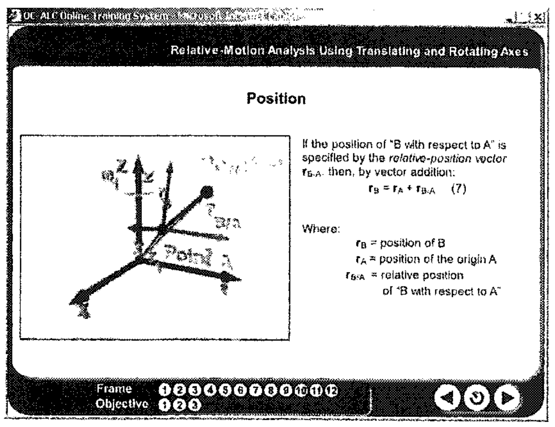
$$dI_{xx} = r_x^2\,dm = (y^2 + z^2)\,dm$$

The moment of inertia I_{xx} for the body is determined by integrating this expression over the entire mass of the body. Hence, the moment of inertia for each of the axes is:

$$I_{xx} = \int_m r_x^2\,dm = \int_m (y^2 + z^2)\,dm \qquad (1a)$$

$$I_{yy} = \int_m r_y^2\,dm = \int_m (x^2 + z^2)\,dm \qquad (1b)$$

$$I_{zz} = \int_m r_z^2\,dm = \int_m (x^2 + y^2)\,dm \qquad (1c)$$


OC-ALC Online Training System
Relative-Motion Analysis Using Translating and Rotating Axes
Frame 1 2 3 4 5 6 7 8 9 10 11 12
Objective 1 2 3

Position

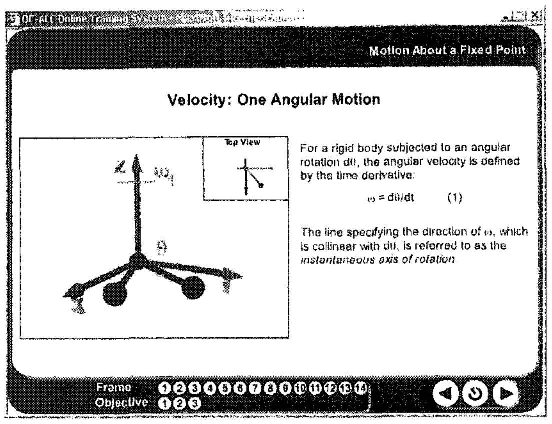
If the position of "B with respect to A" is specified by the *relative-position vector* $r_{B/A}$, then, by vector addition:

$$r_B = r_A + r_{B/A} \qquad (7)$$

Where:

r_B = position of B
r_A = position of the origin A
$r_{B/A}$ = relative position of "B with respect to A"

Figures 5.36a and 5.36b: Sample screens of the *Theory* area with graphics and diagram

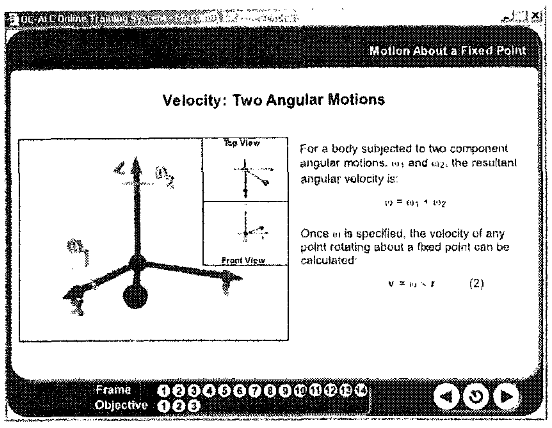

Motion About a Fixed Point
Top View
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Objective 1 2 3

Velocity: One Angular Motion

For a rigid body subjected to an angular rotation $d\theta$, the angular velocity is defined by the time derivative:

$$\omega = d\theta/dt \qquad (1)$$

The line specifying the direction of ω, which is collinear with $d\theta$, is referred to as the *instantaneous axis of rotation*.


Motion About a Fixed Point
Top View
Front View
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Objective 1 2 3

Velocity: Two Angular Motions

For a body subjected to two component angular motions, ω_1 and ω_2, the resultant angular velocity is:

$$\omega = \omega_1 + \omega_2$$

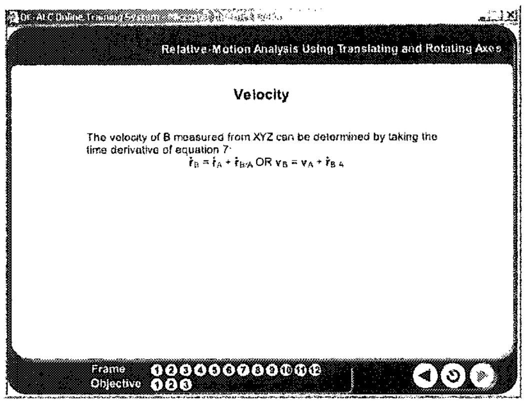
Once ω is specified, the velocity of any point rotating about a fixed point can be calculated:

$$v = \omega \times r \qquad (2)$$

Figures 5.37a and 5.37b: Sample screens of the *Theory* area with animations

In this area, text is another media that is heavily used to present the concepts and equations, as shown in Figure 5.35, which are considered a major part of the engineering and science topics required for particular concepts. In this area, a review of all major mathematical equations used in a particular topic or objective is provided, which could be useful for solving problems in a particular objective as well. This includes explanations and derivations for most of the equations.

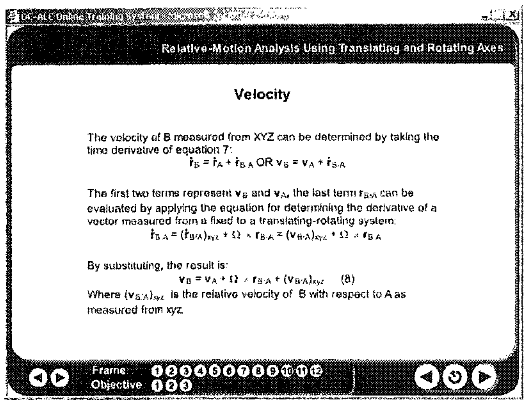
Sound is another major component used in this learning system to help students understand the concepts in the form of audio instruction. Moreover, sound is used to control navigation mechanism of the learning system as well. This detail was already discussed in the development of the EM/C135 System, which is the foundation system of the 3D Dynamics System. Text and sound work related to each other; a particular text is highlighted only when sound explains particular text played. This mechanism helps stress the concepts being explained at a given period of time. Figures 5.38a and 5.38b show sample screens before and after audio instruction is completed.


Relative-Motion Analysis Using Translating and Rotating Axes

Velocity

The velocity of B measured from XYZ can be determined by taking the time derivative of equation 7:

$\dot{\mathbf{r}}_B = \dot{\mathbf{r}}_A + \dot{\mathbf{r}}_{B/A}$ OR $\mathbf{v}_B = \mathbf{v}_A + \dot{\mathbf{r}}_{B/A}$


Relative-Motion Analysis Using Translating and Rotating Axes

Velocity

The velocity of B measured from XYZ can be determined by taking the time derivative of equation 7:

$\dot{\mathbf{r}}_B = \dot{\mathbf{r}}_A + \dot{\mathbf{r}}_{B/A}$ OR $\mathbf{v}_B = \mathbf{v}_A + \dot{\mathbf{r}}_{B/A}$

The first two terms represent $\mathbf{v}_B$ and $\mathbf{v}_A$, the last term $\mathbf{r}_{B/A}$ can be evaluated by applying the equation for determining the derivative of a vector measured from a fixed to a translating-rotating system:

$\dot{\mathbf{r}}_{B/A} = (\dot{\mathbf{r}}_{B/A})_{xyz} + \Omega \times \mathbf{r}_{B/A} = (\mathbf{v}_{B/A})_{xyz} + \Omega \times \mathbf{r}_{B/A}$

By substituting, the result is:

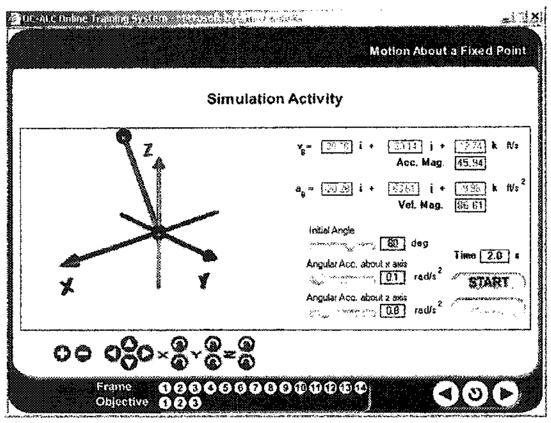
$\mathbf{v}_B = \mathbf{v}_A + \Omega \times \mathbf{r}_{B/A} + (\mathbf{v}_{B/A})_{xyz}$ (8)

Where $(\mathbf{v}_{B/A})_{xyz}$ is the relative velocity of B with respect to A as measured from xyz.

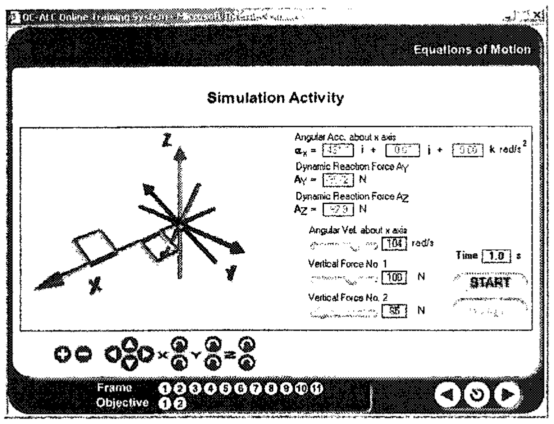
Figures 5.38a and 5.38b: Sample screens before and after audio instruction is completed

2. Simulation Activity

After a theory is introduced, students can then explore the concept behind the theory through a simulation activity before proceeding to the *Problem* area, where students have to apply the theory to solve a realistic problem. However, at anytime while working on the problem, students can invoke a simulation to help solve the problem. A simulation, specifically designed to simulate the problem as closely as possible, provides the opportunity for students to modify parameters, such as mass, velocity, friction, etc. in order to isolate the behavior that exemplifies the concept being illustrated in that particular problem. Students are generally only allowed to modify a maximum of four parameters. Figures 5.39a and 5.39b show sample screens of the *Simulation Activity* area.

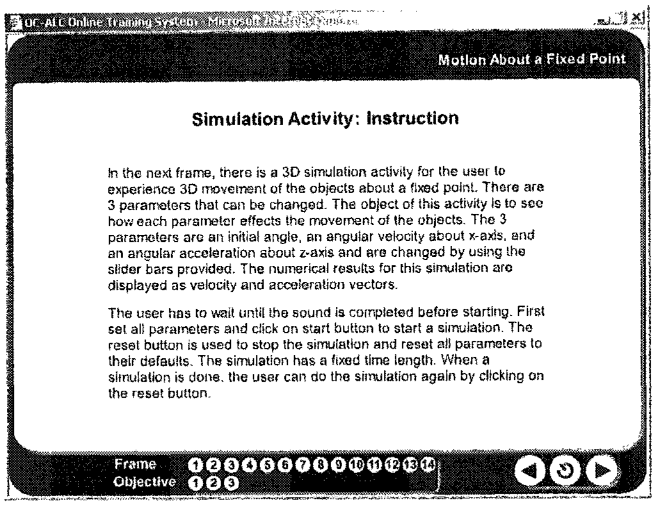


Motion About a Fixed Point
Simulation Activity
Acc. Mag.
Vel. Mag.
Initial Angle
Angular Acc. about x axis
Angular Acc. about z axis
Time
START
Frame
Objective



Figures 5.39a and 5.39b: Sample screens of the *Simulation Activity* area

Before reaching a simulation activity screen, an instruction for each *Simulation Activity* included in each particular topic or objective for that particular simulation activity is also provided to help students utilize the activity effectively. A *Simulation Activity: Instruction* screen includes an experience that students will have (three-dimensional movement of the objects about fixed point, etc.), a simulation activity objective (demonstration how each parameter effects the movement of the objects), how many parameters that can be changed, what those parameters are (initial angle, angular velocity about x-axis, angular acceleration about z-axis, etc.), how to change those parameters (moving sliding bar, entering a value, etc.), and the expected results (velocity, acceleration, etc.). Figure 5.40 illustrates a sample screen of a *Simulation Activity: Instruction*.


Motion About a Fixed Point
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Objective 1 2 3

Simulation Activity: Instruction

In the next frame, there is a 3D simulation activity for the user to experience 3D movement of the objects about a fixed point. There are 3 parameters that can be changed. The object of this activity is to see how each parameter effects the movement of the objects. The 3 parameters are an initial angle, an angular velocity about x-axis, and an angular acceleration about z-axis and are changed by using the slider bars provided. The numerical results for this simulation are displayed as velocity and acceleration vectors.

The user has to wait until the sound is completed before starting. First set all parameters and click on start button to start a simulation. The reset button is used to stop the simulation and reset all parameters to their defaults. The simulation has a fixed time length. When a simulation is done, the user can do the simulation again by clicking on the reset button.

Figure 5.40: Sample screen of the *Simulation Activity: Instruction*

All simulation activities operate identically. Students have to wait until the sound or audio instruction is completed before beginning. Then, students set all parameters and click on the start button to begin a simulation. The reset button is used to stop the simulation and reset all parameters to their default settings. Each simulation has a fixed time length. When a simulation is completed, a user can perform the simulation again by clicking on the reset button. This instruction, which is represented by text and sound or audio instruction, is necessary because although each *Simulation Activity* is identical, each has its own processes.

Complex three-dimensional simulation activities are developed using Lingo scripting language to control and operate entire simulation procedures. Simulation activity screens are divided into three areas; (1) simulated, (2) entering input data and consequence values display areas, and (3) simulated control model area. A simulated area, as shown in Figure 5.41, is where all of the simulation actions, such as translation and rotation of the simulated models or objects, are displayed. The second area (Figure 5.42) includes sliding bars and text fields that are used to control and enter input data, and a consequence values display area that is used to display the numerical outcomes and related values of the simulation activity.

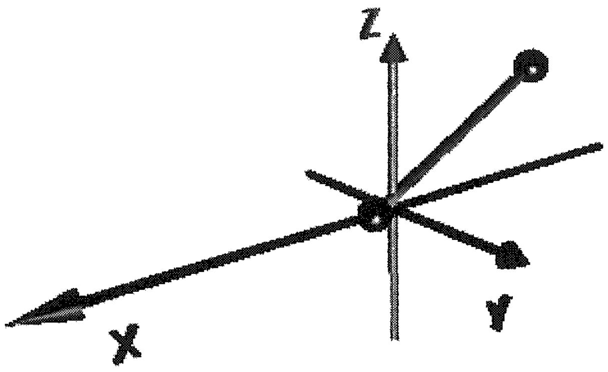
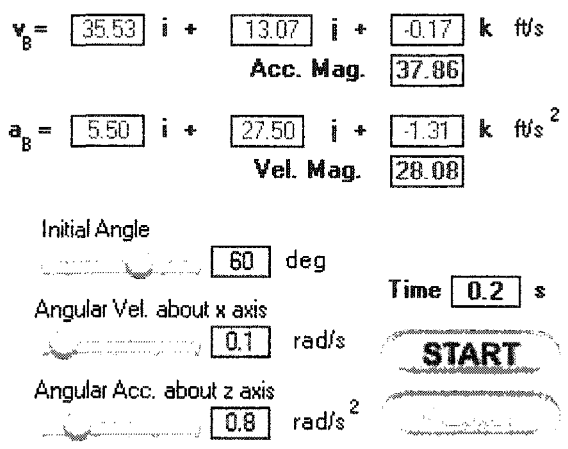


Figure 5.41: Simulated area



v_B = 35.53 i + 13.07 j + -0.17 k ft/s
Acc. Mag. 37.86
a_B = 5.50 i + 27.50 j + -1.31 k ft/s²
Vel. Mag. 28.08
Initial Angle 60 deg
Time 0.2 s
Angular Vel. about x axis 0.1 rad/s
Angular Acc. about z axis 0.8 rad/s²

Figure 5.42: Entering input data and consequence values display areas

The final area, the simulated control area, provides a set of control model buttons that allows users to manipulate the model and view port in order to get the most information out of a simulation, such as the position of the objects during and the end of a simulation. The location of the control model buttons is underneath a simulated area.

These buttons are used to rotate (x, y, z-axes), move (up, down, left, and right directions), maximize, and minimize views of simulated three-dimensional models in order for students to have the best view of model movements and to better understand the theories behind the simulations. Figure 5.43 illustrates a model navigation area provided in the *Simulation Activity* area.

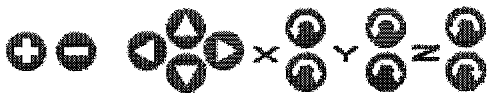
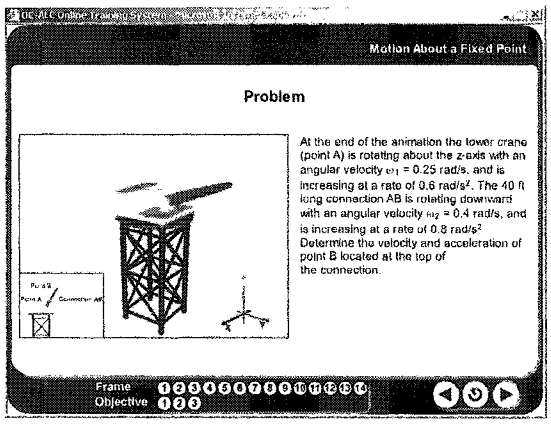


Figure 5.43: Model navigation area for the *Simulation Activity*

After students begin a simulation, all parameters are sent to the function to calculate and come up with expected values to be displayed on the screen and to control all model movements in the simulation. This mechanism is a real-time implementation to make simulations run smoothly with continual movement changes, and to display values continually until a simulation reaches its stopping time or it is terminated by students.

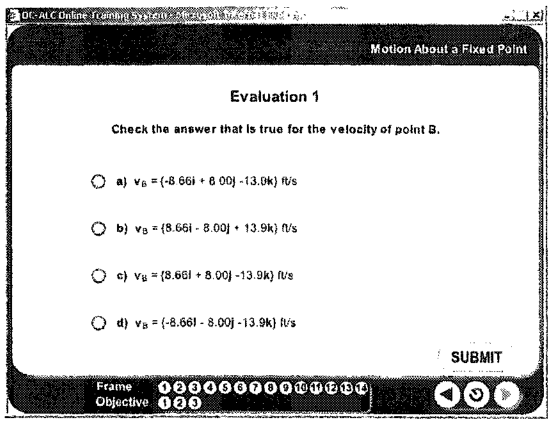
3. Problem

This area presents a detailed problem related to engineering and challenges students to mathematically solve the problem. A particular problem is stated, students are asked to solve the problem. To solve the problem, students have to apply the material presented in the *Theory* area to the problem presented in the *Problem* area. Therefore, if students cannot determine what theory is needed to solve the problem after going through the problem, they can review the theory as many times as they desire. Moreover, after completion of the *Simulation Activity* area, students feel more familiar with the ways to solve the problem. However, if students still have a difficult time solving the problem, they can review the simulation activity based on the parameters introduced in the problem section. Exploring the simulation presented in the *Simulation Activity* area and understanding how the related values are calculated are valuable experiences for students in order to more simply solve the problem and better understand the application of the theory because key aspects are simulated and illustrated during the simulation process. In order to make the problems more captivating and substantial, they are generally realistic, and students are encouraged to approach the problem as if they really had to solve it. The specific details of the problem are presented and an animation is available to help students visualize the problem. This area generally occupies no more than one frame. Figures 5.44a and 5.44b show sample screens of the *Problem* area.


OC-ALC Online Training System
Motion About a Fixed Point
Point B
Point A
Connection AB
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Objective 1 2 3

Problem

At the end of the animation the tower crane (point A) is rotating about the z-axis with an angular velocity ω_1 = 0.25 rad/s. and is increasing at a rate of 0.6 rad/s^2. The 40 ft long connection AB is rotating downward with an angular velocity ω_2 = 0.4 rad/s, and is increasing at a rate of 0.8 rad/s^2 Determine the velocity and acceleration of point B located at the top of the connection.


OC-ALC Online Training System
Motion About a Fixed Point
SUBMIT
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Objective 1 2 3

Evaluation 1

Check the answer that is true for the velocity of point B.

○ a) v_B = {-8.66i + 8.00j -13.9k} ft/s

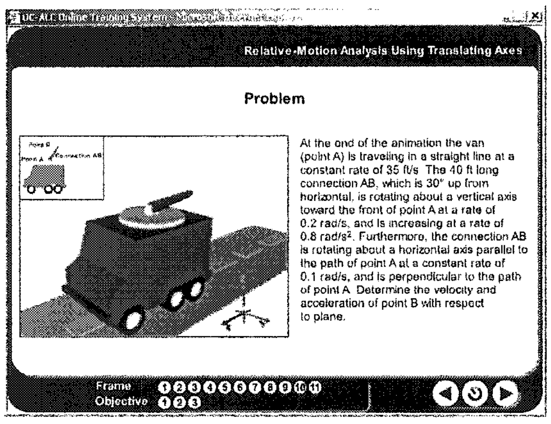
○ b) v_B = {8.66i - 8.00j + 13.9k} ft/s

○ c) v_B = {8.66i + 8.00j -13.9k} ft/s

○ d) v_B = {-8.66i - 8.00j -13.9k} ft/s

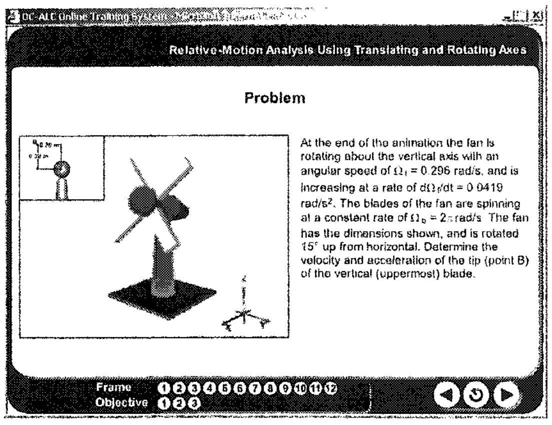
Figures 5.44a and 5.44b: Sample screens of the *Problem* area

The *Problem* area is divided into two sections: problem description (problem introduction) and problem question. Each problem contains a couple of questions that are normally related to each other, such as velocity and acceleration. Problem description is represented by three-dimensional graphics, diagram, and animations, along with text description and audio instruction for students to better understand a particular problem. Figures 5.45a and 5.45b show examples of the problem definition screen.


UC-ALC Online Training System
Relative-Motion Analysis Using Translating Axes
Point B
Point A
Connection AB
Frame 1 2 3 4 5 6 7 8 9 10 11
Objective 1 2 3

Problem

At the end of the animation the van (point A) is traveling in a straight line at a constant rate of 35 ft/s. The 40 ft long connection AB, which is 30° up from horizontal, is rotating about a vertical axis toward the front of point A at a rate of 0.2 rad/s, and is increasing at a rate of 0.8 rad/s^2. Furthermore, the connection AB is rotating about a horizontal axis parallel to the path of point A at a constant rate of 0.1 rad/s, and is perpendicular to the path of point A. Determine the velocity and acceleration of point B with respect to plane.

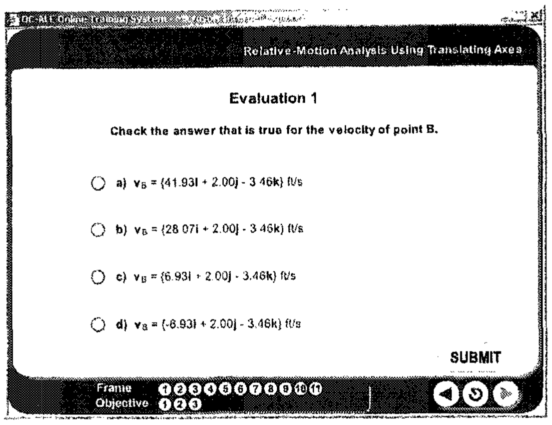

UC-ALC Online Training System
Relative-Motion Analysis Using Translating and Rotating Axes
B
Frame 1 2 3 4 5 6 7 8 9 10 11 12
Objective 1 2 3

Problem

At the end of the animation the fan is rotating about the vertical axis with an angular speed of $\Omega_f = 0.296$ rad/s, and is increasing at a rate of $d\Omega_f/dt = 0.0419$ rad/s^2. The blades of the fan are spinning at a constent rate of $\Omega_b = 2\pi$ rad/s. The fan has the dimensions shown, and is rotated 15° up from horizontal. Determine the velocity and acceleration of the tip (point B) of the vertical (uppermost) blade.

Figures 5.45a and 5.45b: Sample screens of the problem definition

An animation is developed using three-dimensional tools such as Swift 3D and 3ds Max, which are the two applications mainly used in this learning system for three-dimensional modeling. An animation is presented in a perspective view, along with additional views, such as front, side, and top views, to more clearly explain the problem. An animation begins right after students access the *Problem* area, which consists of the problem definition screen and the problem question screens respectively. Additionally, some problem definitions are incorporated with graphics and diagrams as to explain the problem definitions. Reference xyz coordinate system is also provided to clearly explain the direction movement of animation. Text description is another important tool used in conjunction with audio instruction to successfully describe problem definitions. Problem questions are in the form of multiple-choice questions with a single correct answer provided. Figures 5.46a and 5.46b illustrate sample problem question screens.


Relative-Motion Analysis Using Translating Axes

Evaluation 1

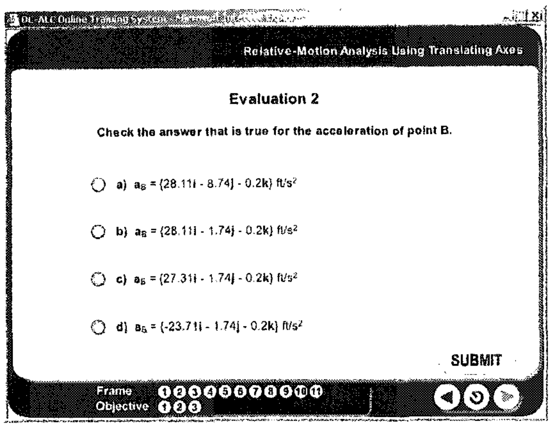
Check the answer that is true for the velocity of point B.

○ a) $v_B = \{41.93i + 2.00j - 3.46k\}$ ft/s

○ b) $v_B = \{28.07i + 2.00j - 3.46k\}$ ft/s

○ c) $v_B = \{6.93i + 2.00j - 3.46k\}$ ft/s

○ d) $v_B = \{-6.93i + 2.00j - 3.46k\}$ ft/s


Relative-Motion Analysis Using Translating Axes

Evaluation 2

Check the answer that is true for the acceleration of point B.

○ a) $a_B = \{28.11i - 8.74j - 0.2k\}$ ft/s^2

○ b) $a_B = \{28.11i - 1.74j - 0.2k\}$ ft/s^2

○ c) $a_B = \{27.31i - 1.74j - 0.2k\}$ ft/s^2

○ d) $a_B = \{-23.71i - 1.74j - 0.2k\}$ ft/s^2

Figures 5.46a and 5.46b: Sample screens of the problem question

Each question provides four answer choices with different feedback for each answer choice after students select the answer and click the submit button. Therefore, regardless of which answer a student selects, the learning system will respond with feedback, whether an answer is correct or not. If an answer is correct, the feedback will congratulate students; however, if it is not correct, the feedback will explain to students why the answer is wrong and also will give students a possible value used in the calculation that might lead to this answer. Moreover, the feedback will give hints as to the correct way to solve the problem, referring to the wrong answers selected by students.

The main reason for using multiple-choice questions to evaluate learners' progress in this research is to imitate the general academic system in the evaluation process. Multiple-choice questions are suitable for web-based evaluations that can be used to effectively track learners' performance. Although all the questions in the 3D Dynamics System require learners to perform computational work to solve problems, multiple-choice questions are still appropriate for the evaluation because the submission of completed computational work is extremely complicated to keep track of through the web. The main objective of these training systems is not just to keep track of learners' performance, but also to provide opportunities for learners to discover the correct solution for a particular question in the evaluation process. Therefore, anytime learners submit incorrect answers, learners can keep trying until correct answers are uncovered. Even though incorrect answers come from learners guessing, learners still get feedback explaining why the answers are mistaken and are given hints to help eventually obtain correct answers.

CHAPTER 6: IMPLEMENTED TECHNOLOGIES AND TECHNOLOGY ASPECTS OF THE SYSTEM

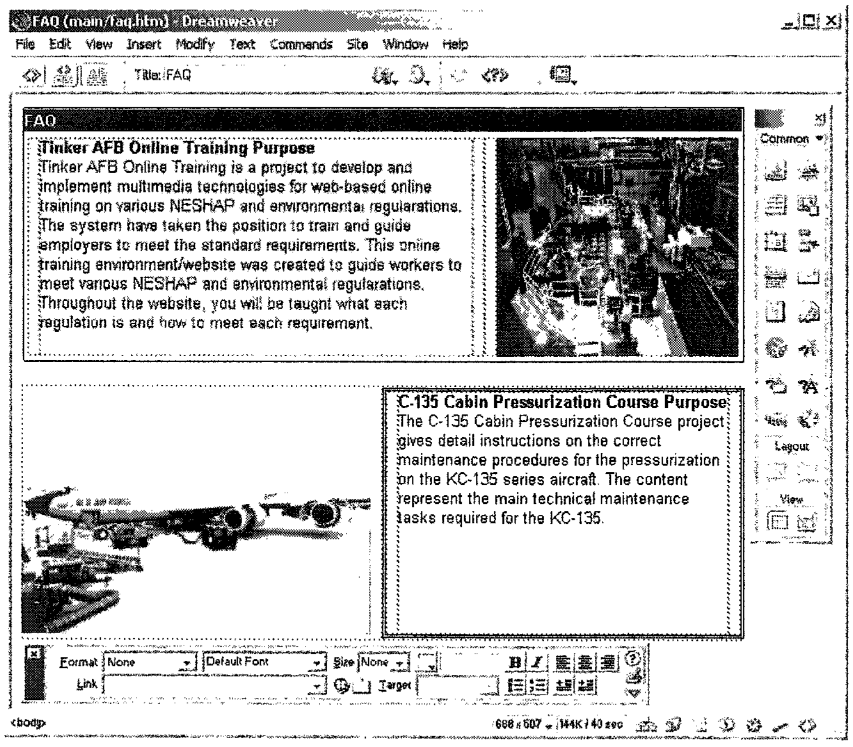
1. Implemented Technologies

There are several Internet and multimedia technologies available in the market that can be used to develop web-based applications. This section discusses the various design tools used for development of web-based training systems presented in this research. These technologies are chosen because they are well defined and commonly used in various applications, including web-based applications. They are employed and smoothly collaborate with each other, achieving all the requirements defined by the objectives of this research. The reasons and benefits for selecting and using these application tools are explained.

1.1 Macromedia Dreamweaver

The majority of the front-end systems consist of static web content and the digital media are embedded in the HTML pages. *Macromedia Dreamweaver* is a visual HTML editor that is used to support the authoring of HTML files. Dreamweaver is a professional tool used for the development and maintenance of web pages and sites [49]. Dreamweaver is a valuable tool since developing static web pages is easier when things are visible to developers. Dreamweaver can create and edit cross platform and cross browser web pages and is equipped with advanced design and layout tools that allow users to manipulate a web page's styles, layouts, images, graphics, and texts.

Dreamweaver is selected because of the extensive support given to developers in strong site management utility that allows fast modification of web pages through the use of templates and frame-based development through a graphical user interface. The advantage of using Dreamweaver is that the clean HTML code and simple JavaScript are generated automatically when development is performed to incorporate special effects. In addition, Dreamweaver has a powerful integrated environment that supports the development of contents using most popular web technologies including DHTML, ASP, JSP, PHP, and XML. Dreamweaver also has built-in functions for inserting applets, VBScript, plug-ins, ActiveX, Flash and Director Shockwave files, and rollover images. Figure 6.1 shows an example of Dreamweaver interface.



Tinker AFB Online Training Purpose
Tinker AFB Online Training is a project to develop and implement multimedia technologies for web-based online training on various NESHAP and environmental regularations. The system have taken the position to train and guide employers to meet the standard requirements. This online training environment/website was created to guide workers to meet various NESHAP and environmental regularations. Throughout the website, you will be taught what each regulation is and how to meet each requirement.

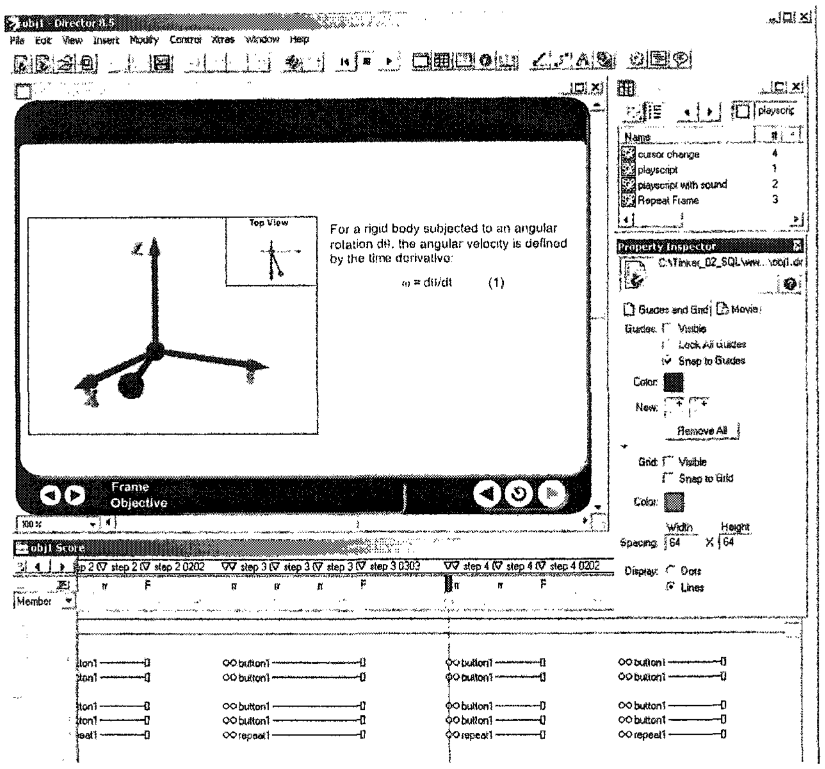
C-135 Cabin Pressurization Course Purpose
The C-135 Cabin Pressurization Course project gives detail instructions on the correct maintenance procedures for the pressurization on the KC-135 series aircraft. The content represent the main technical maintenance tasks required for the KC-135.

Figure 6.1: Example of Dreamweaver interface

1.2 Macromedia Director

The container for all lesson pages was developed with *Macromedia Director*. Director is the foremost multimedia-authoring tool, which is used for creating interactive simulations for the Internet and CD-ROMs [50]. Director is one of the most powerful software tools for authoring multimedia contents with rich interactivity. Director integrates a comprehensive array of multimedia elements to create fast rendering, high-performance, and media rich applications. Multimedia elements such as animation, QuickTime, pixel-and vector-based graphics, text, etc. can be combined in the development environment to create streaming interactive multi-user content.

Director is chosen because of its programming, authoring, streaming, and compression capabilities. Another advantage of using Director is its support of the programming capability known as Lingo scripting language. Lingo is used to design scripts for the navigation system, presentations, and simulations. The scripting flexibility and variable level of control is used to accommodate variety in development by including different styles and templates with very little modification. The ability to communicate with the web-based database and control the media elements through Lingo script without any physical handling helps develop a flexible system that accommodates different responses and requests. A unique feature available in Director that shares commonly used files as an external element provides the graphics elements for the interface in Shockwave. This feature, known as *Shared Cast*, helps in minimizing the effective size of the Shockwave by making the interface graphics external media. This also provides flexibility to make changes easily by making modifications only in one place and not in every Shockwave file. Figure 6.2 illustrates a sample interface of Director.



For a rigid body subjected to an angular rotation dθ, the angular velocity is defined by the time derivative:

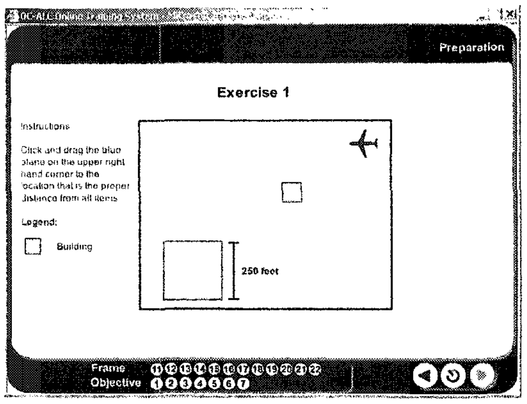
$$\omega = d\theta/dt \quad (1)$$

Figure 6.2: Sample interface of Director

One distinct advantage of Director Shockwave is in its security. Since Shockwave is essentially a compressed compilation of the multimedia and scripting elements of a Director source file, this ensures that the end users cannot decode the download Shockwave file and figure out how to disassemble the file. The attractiveness of this feature is further reinforced with data streaming technology. Streaming technology allows the simulation to be viewed while partially in the process of download. The download time for a Shockwave movie is fast and supports data streaming technology. Streaming is a process whereby a movie will begin playing before the entire movie is downloaded. The movie will start once a minimum amount of media is downloaded, and more will be added as the movie is playing. With streaming technology, users enjoy a shorter download time. This technology allows users to participate in the interaction earlier with less waiting time. Director allows a network operation to begin even though a previous network operation is not complete. This capability is known as background loading and allows files to load while other activities proceed.

In this research, streaming technology is one of the significant features available in Director that is comprehensively utilized to facilitate in downloading simulations. Since most of the simulations are complex, such as the assembly process simulations in the CSD System that incorporates with the complicated three-dimensional assembly model, they consume a large amount of time for entire simulations to be completely downloaded. Streaming technology can reduce download time and also allows users to explore simulations while they are in the downloading process. This technology not only helps in downloading simulations, it is also applicable for animations as well. Some animations are as complex as simulations, such as the assembly animation process in the CSD System, which is mainly composed of an identical three-dimensional assembly model used in assembly process simulations. Streaming technology plays a considerable role in the training systems because it helps make the learning process go smoothly without interruption and wastes less time during animations and simulations.

Another advantage of using the latest version of Director is its capability to support three-dimensional graphics, which allows the delivery of high-performance three-dimensional visualization to the Internet world. This benefit offers a flexible, expandable and robust web-based three-dimensional content creation solution [51]. This feature enhances the capabilities of Director to support interactive three-dimensional graphics as well. Moreover, the produced movie of Director containing three-dimensional content can still be played using browsers with Shockwave plug-in. Three-dimensional objects can be created and controlled using Lingo without any additional computer language or program required. These three-dimensional objects are self-contained inside Director. Moreover, Director allows three-dimensional objects to be imported from other well-known three-dimensional applications such as 3ds Max, Swift 3D, Amapi 3D, Carara, etc. Figures 6.3a and 6.3b illustrate sample simulations developed using Director.



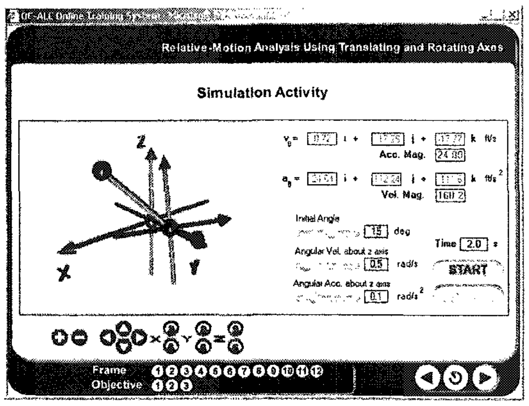
Instructions

Click and drag the blue plane on the upper right hand corner to the location that is the proper distance from all items

Legend:

□ Building

250 feet


OC-ALC Online Training System
Relative-Motion Analysis Using Translating and Rotating Axes
Simulation Activity
Z
X
Y
START
Frame
Objective

Acc. Mag. 24.00

Vel. Mag. 160.2

Initial Angle 15 deg

Angular Vel. about z axis 0.5 rad/s

Angular Acc. about z axis 0.1 rad/s^2

Time 2.0 s

Figures 6.3a and 6.3b: Sample simulations developed using Director

Director's web-based three-dimensional graphics supportive capability is considered the primary feature employed in the training systems associated with this research because it is a unique feature that is mainly used to develop the three-dimensional visualization integrated in the training systems. Three-dimensional visualization that includes graphics, animation, and simulation is a main component of the training systems throughout the learning process. As illustrated in figure 6.3a, it is a two-dimensional simulation in the EM/C-135 System that allow users to move the airplane (blue color) around in a particular area (gray color) and to place it in any location around the building (light blue color) to discover the designated location and the proper distance from all items (airplane, big building at the corner, and small building at the center). In figure 6.3b, three-dimensional simulation is shown. This is one of the complex simulations involved in the 3D Dynamics System. This simulation is developed for users to experience a three-dimensional movement of the objects involved in translating and rotating axes. The object of this simulation is to see how each parameter affects the movement of the objects. There are three parameters that can be changed, including initial angle, an angular velocity about z-axis, and an angular acceleration about z-axis. These parameters can be changed using the slider bars provided, as shown in Figure 6.3b. The numerical results for this simulation are displayed as velocity and acceleration vectors.

1.3 Macromedia Flash

A vector-based animation tool, which is used to develop high quality animations and compress the high-resolution photographs and clip-art images, is developed using *Macromedia Flash.* The compatibility of Shockwave player to run Flash Shockwave files and the wide availability of Flash Shockwave player plug-ins influenced the selection of Flash as an animation tool in this research. In addition, Flash is employed because of its web-based interactivity and ease of use.

Flash is primarily an animation and authoring tool for the web that can also function as a vector graphics editor. It allows both graphics and animations to be created easily through a user-friendly interface while keeping file sizes to a minimum. As maintaining a low file size is the key for quality streaming, the efficient compression capability available with Flash is utilized to develop complex animations at low file sizes. In addition, interactive controls and specified actions can be integrated within the animation without the hassle of dealing with scripting languages. These qualities make Flash a suitable tool for developing animation and interactivity for web pages [52] that include the training systems involved in this research as well. Figure 6.4 illustrates a sample interface of Flash.

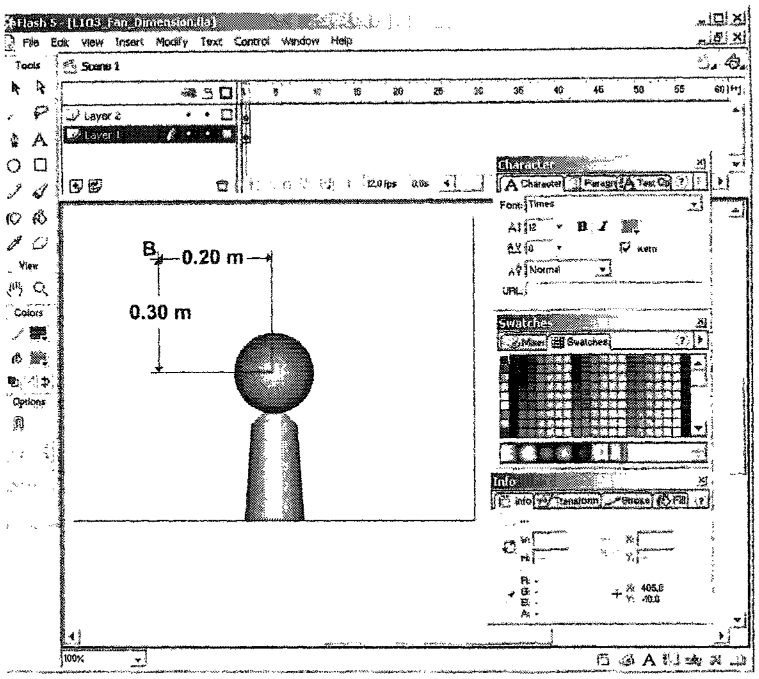
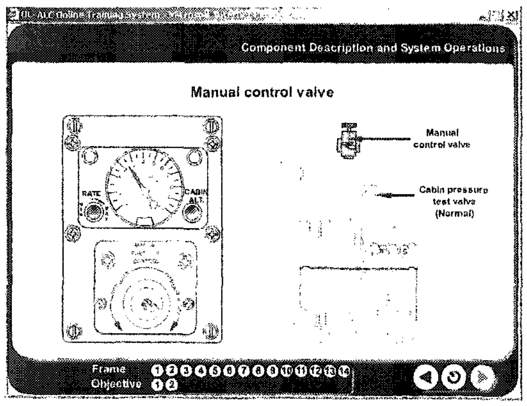
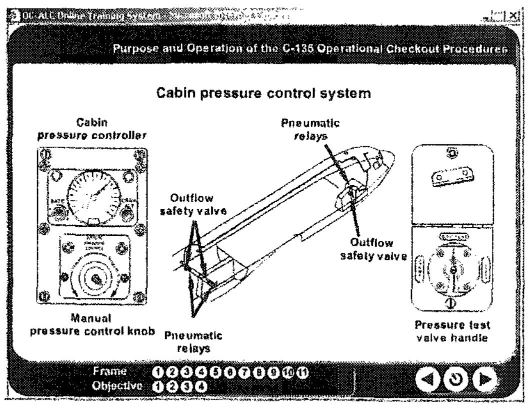

Flash 5 - [L103_Fan_Dimension.fla]
File Edit View Insert Modify Text Control Window Help
Tools
Scene 1
Layer 2
Layer 1
B
0.20 m
0.30 m
View
Colors
Options
Character
Swatches
Info
100%

Figure 6.4: Sample interface of Flash

Inside Flash, the development environment is comparable to that of a movie stage where the different elements are arranged on the time line to determine the order of appearance. The drawing, modifying, and organizing capabilities inside the authoring environment are accessible through its simple visual interface. The layering concept incorporated in Flash is used extensively to create independent entities on different layers so that they can be organized and controlled without affecting other objects. The special features such as masking, and guiding layers are used to give sophisticated effects to the animation without much complexity in programming. These features are broadly used in the development of two-dimensional animations in this research. Figure 6.5a illustrates sample animation that makes use of the masking feature. This feature assists in making some parts of the graphics shown in this animation translucent in order to distinguish a specific part. In this animation, the graphics present a complete component, while the masking feature is of assistance to present only a specific part of the component and make the rest transparent in order for users to focus on that particular part. The guiding layers feature is used to help in the development of animation, as shown in figure 6.5b. In this animation, guiding layers are used to show specific parts in the airplane sequentially. The frame-based development makes Flash a powerful multimedia tool, since it possesses the ability to assign different scripts and dictate the setting independently on different frames. This is another main reason why Flash is used in this research to develop animations. Most animations are controlled using frames, but some complicated animations in particular frames are controlled using scripting beside the frames, as shown in the samples. Figures 6.5a and 6.5b illustrate sample animations developed using Flash.


Component Description and System Operations
Manual control valve
Manual control valve
Cabin pressure test valve (Normal)
RATE
CABIN ALT.
Frame
Objective

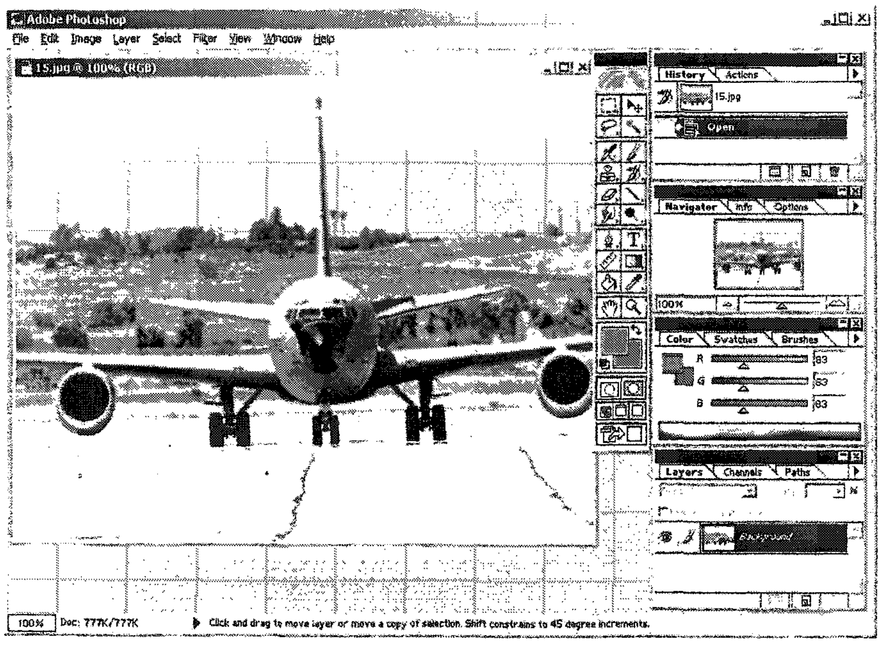

Purpose and Operation of the C-135 Operational Checkout Procedures
Cabin pressure control system
Cabin pressure controller
Pneumatic relays
Outflow safety valve
Outflow safety valve
Manual pressure control knob
Pneumatic relays
Pressure test valve handle
Frame
Objective

Figures 6.5a and 6.5b: Sample animations developed using Flash

In these sample animations, Flash was also used to create the graphics included in these animations. The objective of the first animation (6.5a) is to show a specific part of the component on the right side of the screen following the movement of the knob and the gauge in the control equipment, as shown on the left side of the screen. The next simulation (6.5b) is intended to display a specific part in the airplane following the movement of measurement equipment on the sides of the screen.

1.4 Adobe Photoshop

Adobe Photoshop is used to develop graphics for the front-end interface and to optimize the high-resolution photographs used for training systems involved in this research. The pixel-based environment provides the ability to edit high-resolution images by changing properties such as tint, color, and contrast. Producing sophisticated effects and creating crisp graphics are supreme features available with this tool. Figure 6.6 illustrates a sample interface of Photoshop.



Click and drag to move layer or move a copy of selection. Shift constrains to 45 degree increments.

Figure 6.6: Sample interface of Photoshop

In this research, Photoshop is selected because of its user-friendly interface and the multitude of graphical enhancement styles and effects available. Images and graphics can be created from scratch or imported from other graphics software. Photoshop accepts a wide variety of file import formats, including graphics and images obtained through the use of video and digital cameras, photo CDs, scanners, and video capturing cards. Unlike other existing drawing programs, Photoshop creates and modifies any image by altering the individual pixels [53]. An example of using Photoshop in this research is shown in figure 6.6. In this example, Photoshop is used to edit a photo of the C-135 aircraft. This photo is used in the EM/C135 System. The original size of the photo was reduced and compressed to make the file size smaller, and the format of the photo was also changed to make it compatible and still retain good quality when viewed on the web.

1.5 Sound Forge

Audio is used extensively in the training system to reinforce concepts. As a large number of audio files need to be manipulated, an efficient method to record, digitize, and edit sound is necessary. *Sound Forge* is a Windows-based desktop audio editing and processing tool that is used to edit sound [54]. Sound Forge is used because of its ability to simply cut, copy, and paste sound based on the requirements, to divide the main source files for each frame and easy task. Various sound features such as fading, mixing, and adjusting volume levels are used to edit the media with special effects. Figure 6.7 illustrates a sample interface of Sound Forge.

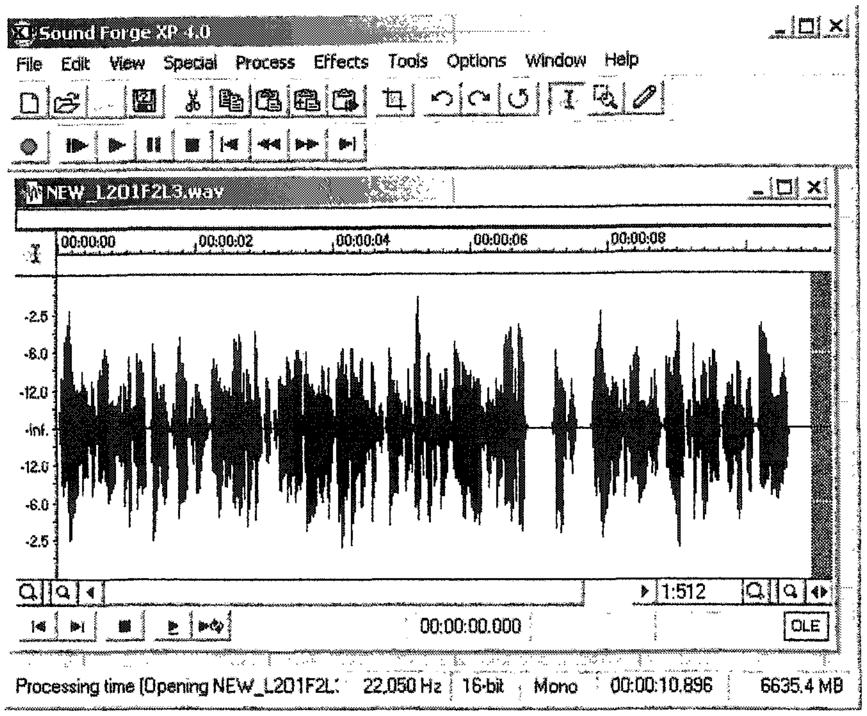
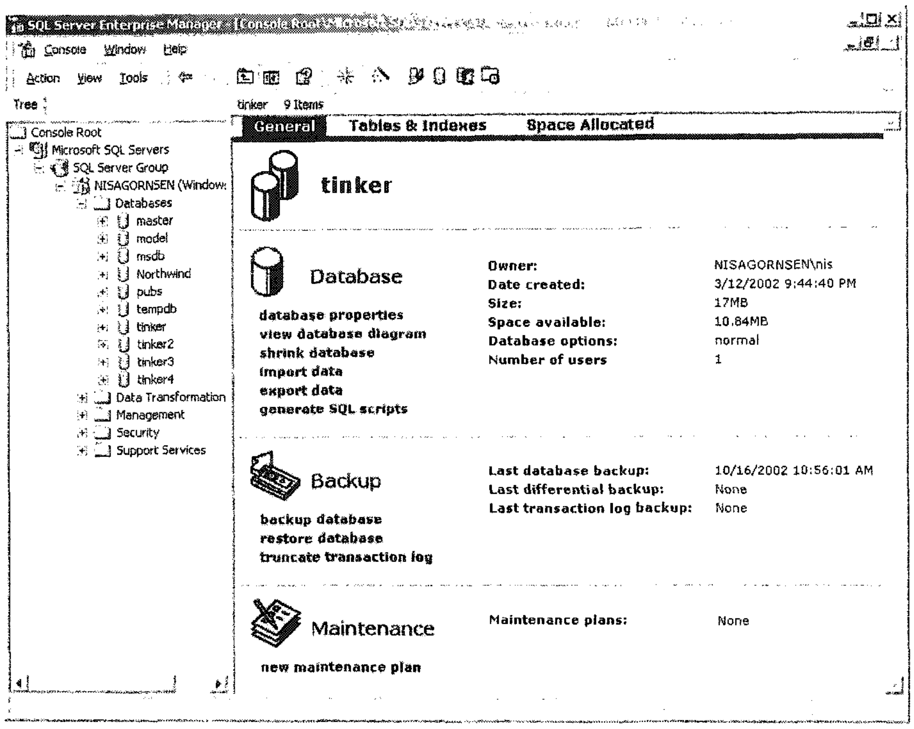

Sound Forge XP 4.0
File Edit View Special Process Effects Tools Options Window Help
NEW_L201F2L3.wav
00:00:00
00:00:02
00:00:04
00:00:06
00:00:08
-2.5
-6.0
-12.0
-Inf.
-12.0
-6.0
-2.5
1:512
00:00:00.000
OLE
Processing time [Opening NEW_L201F2L: 22,050 Hz 16-bit Mono 00:00:10.896 6635.4 MB

Figure 6.7: Sample interface of Sound Forge

1.6 Microsoft SQL Server

The database system in this research is managed using *Microsoft SQL Server*, which is well suited for a web-server-based database because of its simplicity in managing data. The SQL Server database is the ultimate solution for the system to provide storage for a large number of users [55]. It has a capability to effectively support the great volume of concurrently connected users in data processing. These are additional reasons why SQL Server is used in this research, such as its capability to support numerous data stored and simultaneous information retrievals, which are useful for serving a large number of users at Tinker AFB. Another reason for selecting SQL Server as a central place to store data is ASP and SQL, which are used to work with a database to manipulate the data and can smoothly access SQL Server. Moreover, SQL Server database can be easily transformed from Microsoft Access database, which the system database originally incorporated. Figure 6.8 illustrates a sample interface of Microsoft SQL Server.



Owner: NISAGORNSEN\nis
Date created: 3/12/2002 9:44:40 PM
Size: 17MB
Space available: 10.84MB
Database options: normal
Number of users 1

Last database backup: 10/16/2002 10:56:01 AM
Last differential backup: None
Last transaction log backup: None

Maintenance plans: None

Figure 6.8: Sample interface of Microsoft SQL Server

SQL Server is a fully web-enabled database and data analysis package. SQL Server has an inbuilt data warehousing system that can integrate, consolidate, and summarize information through large, central data stores where information is collected. It supports all of the general requirements of a database, such as integrity (e.g. information retrievals from Tinker or non-Tinker users that are received from appropriate locations in the database are accurate), concurrent access (e.g. several users at Tinker AFB taking the training at the same time), and security (e.g. unassigned users, non-Tinker users, and guest users are not allowed to access the training systems or the restricted parts of the training systems).

SQL Server is also scalable with its dynamic self-management features such as dynamic space management that automatically grow or shrinks the database size, and dynamic memory allocation for faster database performance. This feature is significant in this research because information inquiries always occur due to a number of users and a database has to adjust depending upon connected users in order to better serve users. It has a wide array of replication options, which allows users to modify data independently and synchronize data into a singular uniform product. SQL Server also provides tools to monitor the circumstances under which automated management tasks occur, such as changes in data files and log files. Developers, in turn, can troubleshoot problems by capturing these events on a production system and replaying them on a test system.

SQL Server's performance is also optimized for complex queries. Its seamless integration with Windows 2000 Server provides security for use in web-based applications. This capability provides the highest level of security available in the industry, taking advantage of the integrated security of Windows 2000 Server.

1.7 Structure Query Language (SQL)

SQL stands for Structured Query Language and is a standardized language used to communicate with a database. SQL is a user-friendly powerful language for querying a database that utilizes a combination of relational algebra and relational calculus. Because of this, SQL works only with relational databases such as Microsoft Access, Microsoft SQL Server, and Oracle. These databases have their own proprietary languages to interact within their programs. Besides performing as a query language, SQL can define data structures, manage database objects, manipulate data and manage data transactions and can also manage database security. These queries include obtaining information and updating data on a database. SQL queries take the form of a command language that lets the user select, insert, update, find the location of data, and so forth. Figure 6.9 gives a picture illustration of the SQL usage.

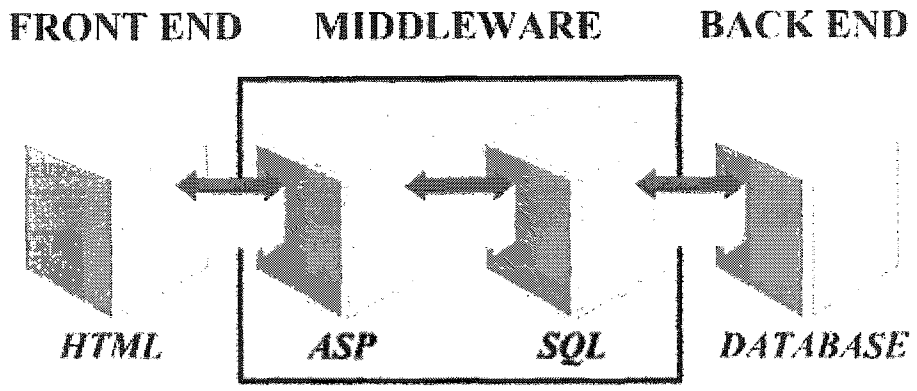


Figure 6.9: Illustration of SQL usage

Queries using SQL appear in several forms including commands such as SELECT, INSERT, UPDATE and DELETE, and clauses such as WHERE and ORDER BY. Commands and clauses are also used extensively in this research. SELECT is used most frequently to retrieve information, such as users' names and types. This information retrieval can be from one or more tables in the database. INSERT is used to insert (add) information such as users' latest evaluation scores and users recently added to the system into a database table. To update (modify) existing data in a table, such as users' addresses and system preferences the UPDATE statement is used. To delete data (remove), such as users and lessons from a table, the DELETE statement is used. WHERE is used to specify the location of information such as the user names' row in the user information table. The ORDER BY clause is used to explicitly sort data retrieved, such as after retrieving all usernames in the database and then sorting them alphabetically (e.g. Ant, Bee, and Cat). There are several levels of SQL commands. They can be classified as the data definition language (DDL), data manipulation language (DML), and data control language (DCL) [56]. DDL contains commands used in defining a database such as create, alter, and establishing constraints. In other words, it is used to create and destroy databases and database objects. After the database structure is defined with DDL, database administrators and users can utilize DML to insert, retrieve and modify the data contained within it. The DML commands are used in maintenance and in queries to the database such as insert, update, and query. Finally, DCL commands are used in controlling the database, causing such effects as privileges and committing of data.

In the system, SQL statements are embedded in ASP scripts to access the SQL database. The scores of the evaluation are sent from Director to an ASP file. Inside the ASP file, SQL statements, acting as information transferring controllers, receive and send information to the database. This information is stored in the database and can be retrieved from the database when it is required. The results are presented in text format or HTML format after the ASP scripts are interpreted.

1.8 ASP Scripts

To track the user and update the database, the server-side *ASP* (Active Server Pages) programming language is used. The results of this are presented in dynamically generated HTML pages. This language is executed on the server in response to requests made by the user or browser. ASP interfaces well with all web-based technologies. ASP technology is used extensively in this research due to several reasons, including its low learning curve, simple integration of ASP technology to Windows 2000 Server environment, easy database communication, stability and rapid development of the technology. Sample ASP script is shown as follow:

```
<%
username=request.querystring("userid")
If username NOT empty then
     Response.write("My username is " & username & ".")
End if
%>
```

ASP is a server-side scripting environment that can be used to create dynamic web pages and build powerful web applications [58]. An ASP is a HTML page that contains scripts that are dynamically processed on the web server before the ASP page is sent to a client machine. When the server receives a request from an ASP file, it processes server-side scripts contained in the file to build the web page that is sent to the browser. The output of such a process is in the form of a static web page (as a normal HTML page). This indicates performance of ASP as browser- and operating system-independent. Therefore, the web page's dependency on the client's browser type is diminished. Since users might use different types of browsers (Netscape Navigator and Microsoft Internet Explorer) on different types of computers (PC and Macintosh), it is beneficial that ASP is chosen as the environment for developing web content. Among many competing technologies used to create dynamic web contents, such as PHP (Hypertext Preprocessor), CGI (Common Gateway Interface), and JSP (Java Server Pages), ASP shows some distinctive advantages, such as high running efficiencies on the web server due to good reusability of server threads, and the quality of being easy to learn.

ASP files contain scripts written in languages that the web server understands such as VBScript and JavaScript, which are commonly used for this purpose. In this system, ASP is based on VBScript to generate HTML pages dynamically, process client's requests, and to access the online databases. Another benefit of the ASP technology is security. As the ASP scripts are processed on the server, users on client machines cannot view the server-side scripts. The users only see the output of the generated static HTML file, and have a harder time to compromise the security of the server. In addition, ASP makes it easier to protect copyright.

An example of the use of ASP in this research is shown at the beginning of the training systems. When users first access the training systems, usernames and passwords are required for users to identify themselves. After users enter usernames and passwords, ASP files on the server will be executed, then will access the database and retrieve the necessary data for users who exist. Then ASP embeds the process results in HTML to generate a customized web page for individual Tinker training users. The result page depends on a user's identity as retrieved from the database, taking into account the type of user and how many lessons the user has completed.

1.9 VBScript

VBScript is the Microsoft scripting language that was created as an alternative to Netscape's JavaScript. VBScript was designed to work on Microsoft's Internet Explorer and includes advanced programming capabilities such as ActiveX controls that allow more integration into other Microsoft-based technologies. Although VBScript was designed for the Internet, Netscape (version 4 and below) does not support VBScript. Therefore, use of VBScript has to be restricted to the Internet Explorer web browser only, which poses a problem to web developers [60].

In this system, VBScript is used extensively in ASP pages. Since ASP pages are processed on the server, use of VBScript does not incur browser incompatibility. Using VBScript solely as server-side script is to avoid causing browser incompatibility or forcing the users to use certain browsers. VBScript is used because it is more descriptive and easier to learn compared to JavaScript. Furthermore, plenty of reference and learning materials are available on the web. Simple VBScript script can be written as follows:

```
<script language= "VBScript">
Sub VBMsg_OnClick
MsgBox "This is a message box generated by VB Script"
End Sub
</script>
```

1.10 JavaScript

JavaScript was developed by Netscape as an interpreted programming language to enhance web pages' performance. JavaScript is used to add interactivity to the HTML pages [61]. This interactivity includes dynamically generating content in HTML pages, controlling the browser, and transferring information among HTML pages. The page content can be based on user input, which makes HTML pages distinct from traditional books. This scripting language existed earlier than Microsoft's VBScript and was created to add functionality and interactivity to static web pages. Both Microsoft's Internet Explorer and Netscape's Navigator recognize JavaScript. Although JavaScript can also be used as server-side script, in this system, it is mainly used in static web pages as client-side script to follow directions such as opening a pop-up window and displaying mouse-rollover effects on text and graphics. This is because of its cross-browser features. JavaScript can be conveniently embedded into an HTML file, as shown in the following sample, which generates a message box with texts inside:

```
<script language= "JavaScript">
Function JSMsg()
{
Alert("This is a message box generated by JavaScript")
}
</script>
```

1.11 Swift3D

As impressive as Flash animation is, the integration of three-dimensional objects and environments into a design can be even more exciting. *Swift3D* is a simple yet powerful tool for the creation of three-dimensional animations to be imported as Flash movies, which are in the format of *swf* files, for use in various multimedia technologies such as Flash and Director. Swift3D is an intuitive three-dimensional application for creating, editing, and animating three-dimensional objects that are compatible with the Flash application [62]. This application is an excellent solution for the multimedia developer looking to add from basic to intermediate levels of three-dimension to the mix, which is the main reason this tool is used in this research. Figure 6.10 illustrates a sample interface of Swift3D.

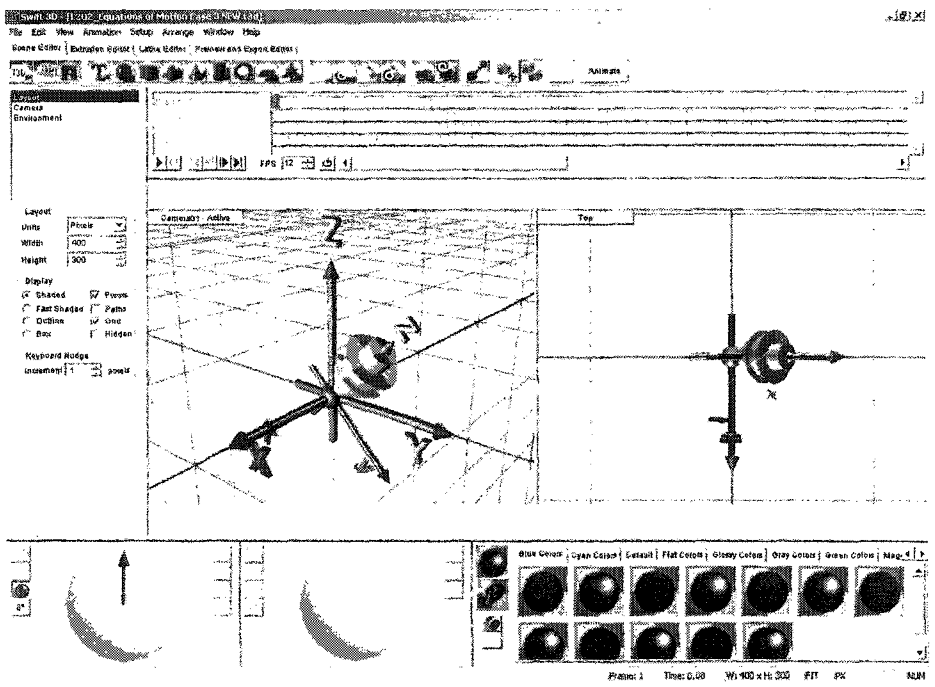
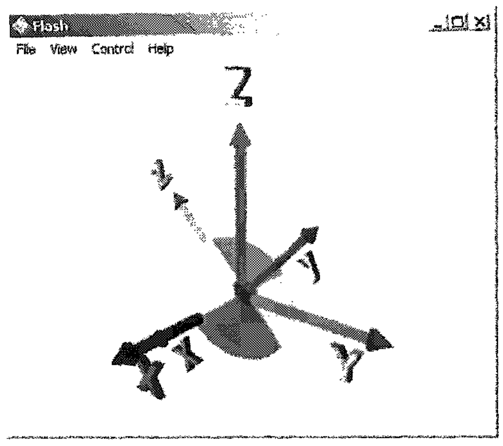
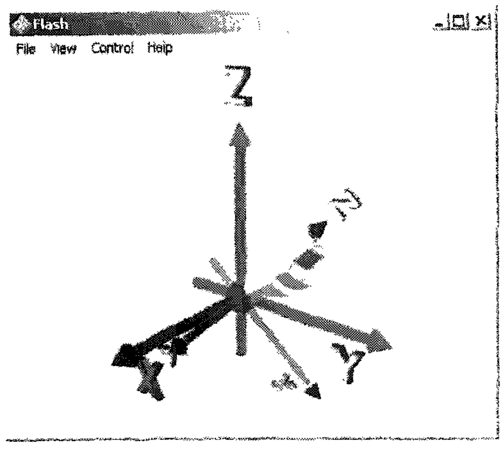


Figure 6.10: Sample interface of Swift3D

Most three-dimensional applications have a steep learning curve, but Swift 3D is quite simple to grasp. The easy-to-understand manual provides a great introduction to three dimensions for multimedia designers who have not attempted to use a three-dimensional application before to be able to begin creating meaningful objects in a short period of time. Swift 3D is designed to create three-dimensional content for the Flash application with its high usability even without previous experience, distinguishing it from other applications. It is easy and useful in creating excellent web content including web design and web-based applications. While not as refined as some software applications out there, the interface is laid out well and fairly intuitive. Its user-friendly interface allows the user to concentrate on the function of the software instead of figuring out the position of the buttons. In addition to the stand-alone version, the Swift 3D plug-in version is also available for many high-end three-dimensional applications, allowing the user to create high-level three-dimensional models and animations like *3ds Max.* While the stand-alone version is a quality product, it does not touch the three-dimensional creation capabilities of dedicated professional three-dimensional applications. Figures 6.11a and 6.11b illustrate sample three-dimensional model and animation developed using Swift3D.





Figures 6.11a and 6.11b: Sample model and animation developed using Swift3D

In these samples, Swift 3D was used to develop intermediate level three-dimensional models and animations for the 3D Dynamics System. The first animation (figure 6.11a) is used to show the rotational motion of the object (cylinder) around the Y-axis; the object's center of rotation is located at the center point of the XYZ coordinate system. The object itself has no rotation movement. In the second animation (figure 6.11b), the object (gyroscope) not only rotates around the Y-axis, it also spins around itself, which is the z-axis in this particular sample. The object's center of rotation for both rotational motions is located at the center point of the XYZ-coordinate system as well. These animations help users better understand the topic of Equation of Motion and enable users to analyze the variety movements that are possible in three dimensions.

1.12 3ds Max

3ds Max (previously called 3D Studio Max) is a multifunction, three-dimensional modeling, rendering, and animation tool [65]. It has been used for many purposes, including special effects in films and advertisements and for complex objects in virtual environments. This application is a complete solution for creating a three-dimensional world. It is also the world's most widely used professional three-dimensional modeling, animation and rendering solution. It contains all the essential tools required for creating eye-catching animation, cutting-edge games, and distinctive design visualizations. It is built for the creative professional who is advancing into the world of three-dimension. Figure 6.12 illustrates a sample interface of 3ds Max.

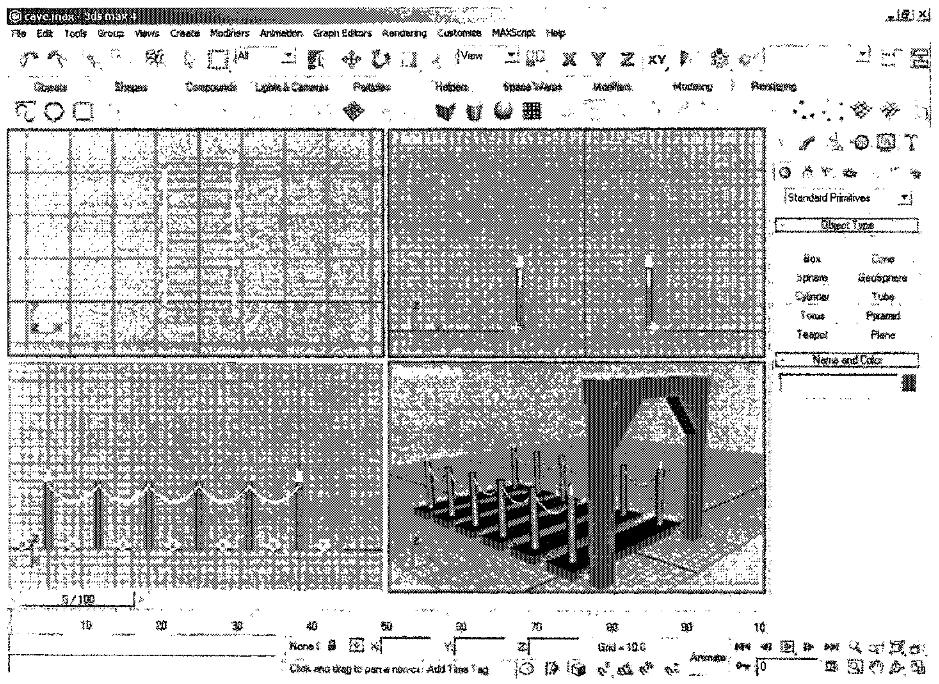
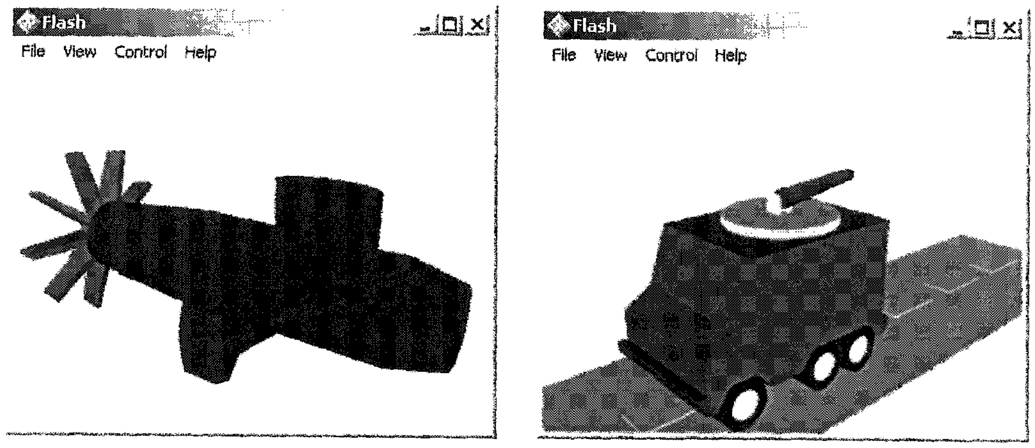


Figure 6.12: Sample interface of 3ds Max

The decision of employing 3ds Max together with the Swift3D for this research was made because of its ability to develop high-level three-dimensional models and animations. In addition, 3ds Max supports many export formats including w3d, which is the only three-dimensional model format supported by Director. This makes the cross platform application easy. Moreover, three-dimensional models and animations developed using 3ds Max can be exported as flash files (*swf*) using the Swift 3D plug-in available for 3ds Max and LightWave 3D only [66, 67, 68]. The decision to use 3ds Max over LightWave 3D is because although both software have the same abilities required for this research, the cost of the 3ds Max software is cheaper and the learning resources need for 3ds Max are available in the EML Lab. With the capability to convert three-dimensional scenes to vectors in Swift 3D plug-in, it enables high-end content developed using 3ds Max to be converted into low-bandwidth *swf* files of exceptional quality that stream seamlessly over the web [67]. This ability provides the flexibility needed to expand the horizons of the high-level three-dimensional content that can be experienced through the web. This Swift 3D plug-in supplies the most advanced capabilities for vector output to the *swf* file format. The vector rendering technology in Swift 3D plug-in exports three-dimensional scenes with supreme accuracy, while staying true to colors, lighting schemes, camera views, and animations. It is able to handle models and objects containing over 200,000 polygons, including complex intersecting and self-intersecting objects [67]. This plug-in picks up where others drop off. Figures 6.13a and 6.13b illustrate sample three-dimensional model and animation developed using 3ds Max.



Figures 6.13a and 6.13b: Sample model and animation developed using 3ds Max

In these samples, 3ds Max was used to develop animations for the 3D Dynamics System. These animations are used alongside text descriptions of the problems in the *Problem Area*. With the use of animations, students can understand problems more clearly. The first animation (figure 6.13a) shows the motion of a submarine in three dimensions Students are asked to determine the inertial properties and the angular momentum of the coning tower about the center of gravity of the submarine when the movement commences. In the second animation (figure 6.13b), students are asked to determine the velocity and acceleration of the top point of rotating connection on the top of the van with respect to plane. In this animation, the van moves in a straight line while the connection rotates simultaneously.

2. Technology Aspects of the System

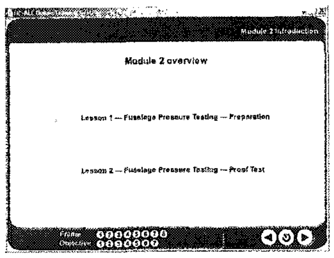
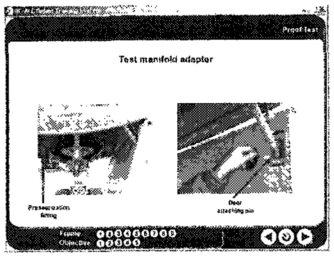
A significant advantage of using the web for engineering education and technical training is the availability of a wide range of media that can be used effectively for presentation. Information delivery is done by integrating a variety of multimedia components, such as videos, animations, sounds, and simulations with the standard text and graphics. These electronic media types are combined to match nicely with many of the aspects of technical training. Additionally, the opportunity for users to explore and learn effectively through an investigative process is provided using these technologies.

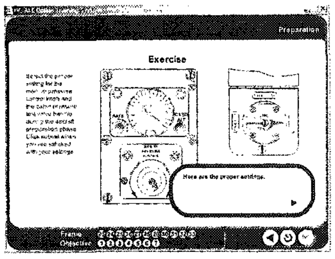
As technical and engineering training require a high level of interactivity, a double stage container system is used in delivering the information. The first level container is the Internet browser. The browser is used because of its ability to interface the client within the World Wide Web with a simple connection. Further, the main difficulty with any web-based delivery is the creation of content material with a simple navigation system and a small file size. Therefore, in order to maximize the benefits of electronic media, a secondary container, Shockwave, was used. The technology and techniques discussed in this section explain how this media was actually implemented in the lessons.

2.1 Text

Text is used to provide brief instruction or summaries of concepts in each frame so that users are not overloaded and tired with too much content. An instructional tone that emphasizes the information is used to keep the users' attention. Anti-aliased text that combines with the background color without any sharp edges is used to make the presentation environment attractive. An aspect, considered before incorporating this media element in the system, is the ease of authoring in *Director* since text is directly supported. Text elements are widely used to summarize learning material since an internal text editor, easily accessible through a graphical user interface, is available.

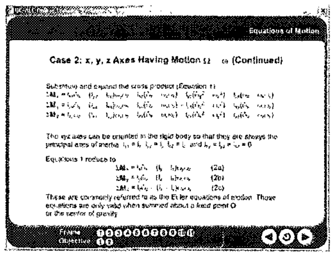
Besides mainly using text for brief summaries of concepts, text is used for other several purposes throughout the systems. Text serves as labels for subjects as shown in figure 6.14a and other elements included in frames such as graphics and animations (figure 6.14b). Text is employed as feedback for evaluated questions, and also some animation and simulation activities (figure 6.14c). This use assists students and users to improve learning process with correct information explanations.

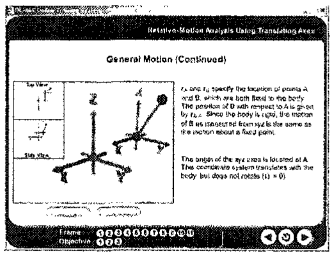
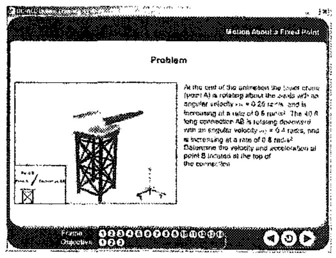

Module 2 Introduction
Module 2 overview
Lesson 1 — Fuselage Pressure Testing — Preparation
Lesson 2 — Fuselage Pressure Testing — Proof Test
Frame
Objective




Figures 6.14a, 6.14b, and 6.14c: Sample uses of text for labels and feedback

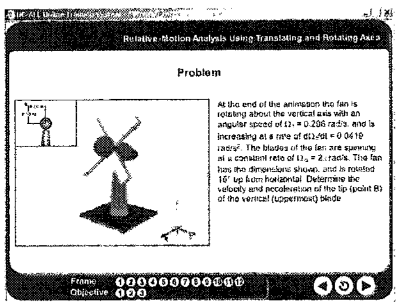
Text is used to present mathematical equations and mathematical numbers in order to for users to effectively attain the concept of *Three-dimensional Rigid Body Motion* in the 3D Dynamics System as illustrated in Figure 6.15a. Text is also applied as labels to define views of animations such as top view and front view as shown in figure 6.15b. Additionally, text provides details about animations such as located positions for point A and point B (figure 6.15c).




Relative-Motion Analysis Using Translating Axes
General Motion (Continued)
Top View
Side View
Frame
Objective


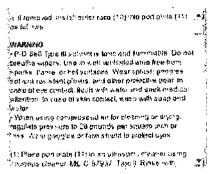
Figures 6.15a, 6.15b, and 6.15c: Sample uses of text for equations, labels, and locations

Figure 6.16a illustrates text clarifies dimensions of objects in animations such as length and width of fan blade. Text instruction description explains step-by-step assembly process for the *Aircraft Part and Maintenance Process* in the CSD System as shown in figure 6.16b. Part names and IDs in the *left menu area* are displayed using text as well (figure 6.16c).


Relative Motion Analysis Using Translating and Rotating Axes

Problem

At the end of the animation the fan is rotating about the vertical axis with an angular speed of Ω_1 = 0.206 rad/s, and is increasing at a rate of $d\Omega_1/dt$ = 0.0419 rad/s². The blades of the fan are spinning at a constant rate of Ω_2 = 2 rad/s. The fan has the dimensions shown, and is rotated 15° up from horizontal. Determine the velocity and acceleration of the tip (point B) of the vertical (uppermost) blade

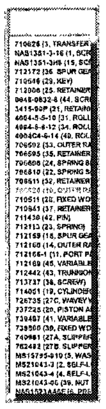

If removed, install outer race (10) into port plate (11) as follows.

WARNING

P-D-680 Type III solvent is toxic and flammable. Do not breathe vapors. Use in well ventilated area free from sparks, flame, or hot surfaces. Wear splash goggles, solvent resistant gloves, and other protective gear. In case of eye contact, flush with water and seek medical attention. In case of skin contact, wash with soap and water.

When using compressed air for cleaning or drying, regulate pressure to 29 pounds per square inch or less. Wear goggles or face shield to protect eyes.

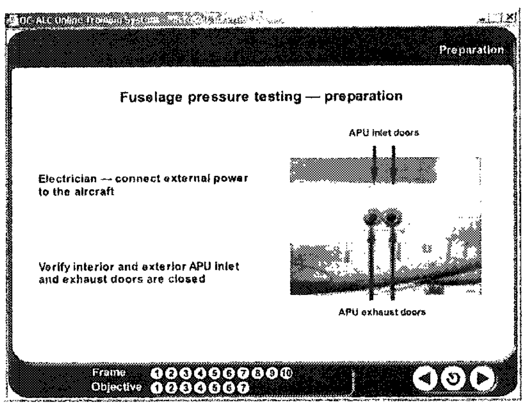
(1) Place port plate (11) in an ultrasonic cleaner using aqueous cleaner MIL-C-87937 Type II. Rinse with


710626 (3, TRANSFER
NAS1351-3-16 (1, SCR
NAS1351-3H8 (15, SCR
712172 (36, SPUR GEA
710546 (29, KEY)
712006 (25, RETAINER
0848-0832-8 (44, SCR
3415-98P (21, RETAIN
4064-5-5-10 (31, ROLL
4064-5-6-12 (34, ROLL
4064C4-6-14 (40, ROLL
706592 (33, OUTER RA
706595 (35, RETAINER
706606 (24, SPRING S
706610 (22, SPRING S
706611 (32, RETAINER
[illegible]
710511 (28, FIXED WO
710541 (37, RETAINER
711430 (42, PIN)
712113 (23, SPRING)
712159 (16, SPUR GEA
712160 (14, OUTER RA
712166-1 (11, PORT P
712169 (45, VARIABLE
712442 (43, TRUNNION
713737 (38, SCREW)
714051 (19, CYLINDER
726735 (27C, WAVEY W
737225 (20, PISTON A
739487 (41, VARIABLE
739500 (30, FIXED WO
740961 (27A, SLIPPER
762492 (27B, SLIPPER
MS15795-810 (5, WAS
MS21043-3 (2, SELF-L
MS21043-4 (4, SELF-L
MS21043-06 (39, NUT
NAS1523AA6E (6, PRE

Figures 6.16a, 6.16b, and 6.16c: Sample texts for dimensions, instruction, and names

2.2 Graphics

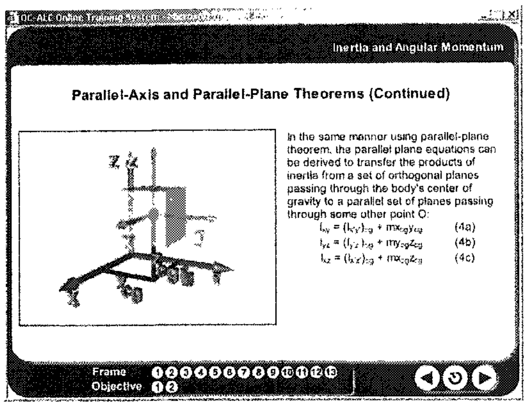
Graphics are used in the system to enhance users' learning experience through visual motivation, which is very helpful to show technical aspects and complex concepts of the training. Graphics provide visual impact as well as information to users and students. Since *graphics is worth more than a thousand words*, graphical illustrations are commonly used in the training system. The graphics used are schematic diagrams developed with graphic authoring tools, images taken from clipart, and high-resolution photographs of people, processes, and equipment. Most of the graphics used are edited in *Photoshop*, and then imported into *Flash* or *Director*. Figures 6.17a and 6.17b are examples of graphics.



Fuselage pressure testing — preparation

Electrician — connect external power to the aircraft

Verify interior and exterior APU inlet and exhaust doors are closed


OC-ALC Online Training System
Inertia and Angular Momentum
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13
Objective 1 2

Parallel-Axis and Parallel-Plane Theorems (Continued)

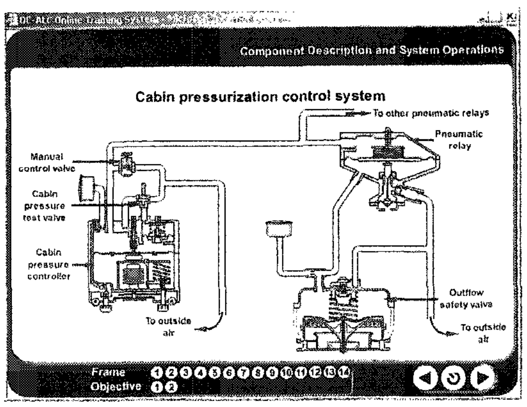
In the same manner using parallel-plane theorem, the parallel plane equations can be derived to transfer the products of inertia from a set of orthogonal planes passing through the body's center of gravity to a parallel set of planes passing through some other point O:

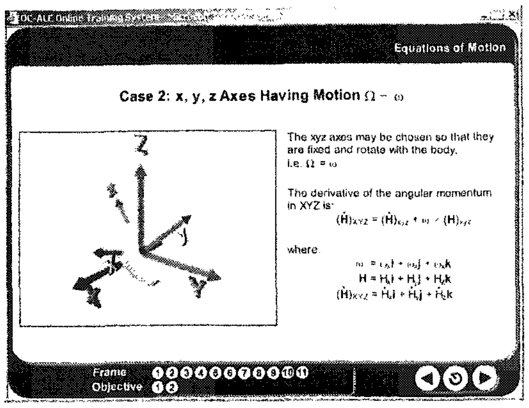
$$I_{xy} = (I_{x'y'})_{cg} + m x_{cg} y_{cg} \quad (4a)$$
$$I_{yz} = (I_{y'z'})_{cg} + m y_{cg} z_{cg} \quad (4b)$$
$$I_{xz} = (I_{x'z'})_{cg} + m x_{cg} z_{cg} \quad (4c)$$

Figures 6.17a and 6.17b: Examples of graphics

2.3 Animation

Animation is used to introduce topics and to explain basic principles that are relevant to the users. Animation is integrated along with the other elements to improve the value of the educational experience. Animation is a key media that can raise the interest and understanding of the user since static environments with just text are uninteresting. Animation is used to take advantage of and to focus users' attention to support a stimulated learning process. Also, animation is used to stage story-like presentations, show processes, and clearly explain complicated concepts such as technical ideas that need to be represented over a period of time. *Flash* is mainly used to develop two-dimensional animation that can be controlled inside Director using Lingo scripting language while *Swift3D* and *3ds Max* are major tools for the creation of three-dimensional animations. These technologies are already discussed in the first section of this chapter. Examples of animations integrated with other presentation material in the environment are shown in Figures 6.18a and 6.18b.


OC-ALC Online Training System
Component Description and System Operations
Cabin pressurization control system
To other pneumatic relays
Pneumatic relay
Manual control valve
Cabin pressure test valve
Cabin pressure controller
To outside air
Outflow safety valve
To outside air
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Objective 1 2



Case 2: x, y, z Axes Having Motion $\Omega = \omega$

The xyz axes may be chosen so that they are fixed and rotate with the body.
i.e. $\Omega = \omega$

The derivative of the angular momentum in XYZ is:

$$(\dot{H})_{XYZ} = (\dot{H})_{xyz} + \omega \times (H)_{xyz}$$

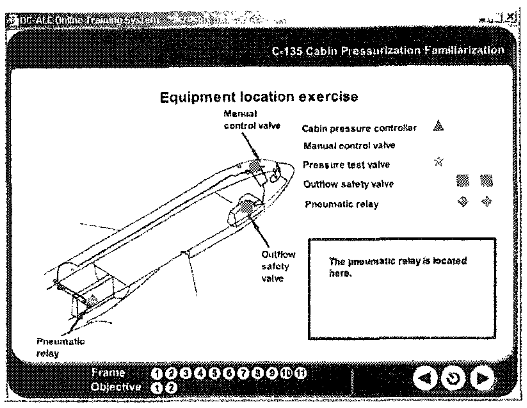
where:

$$\omega = \omega_x i + \omega_y j + \omega_z k$$
$$H = H_x i + H_y j + H_z k$$
$$(\dot{H})_{xyz} = \dot{H}_x i + \dot{H}_y j + \dot{H}_z k$$

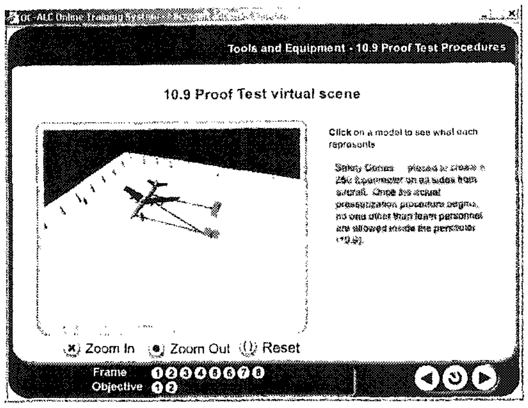
Figures 6.18a and 6.18b: Examples of animation

2.4 Simulation

Simulation, which is considered an informative and attractive element in the learning process, is used periodically to bring life to many of the basic concepts and to illustrate technical aspects by providing users with hands-on training through an interactive exploration process along with the presentation material. The main purpose of employing simulation in training is that it explains technical technologies by imitating real life environments and situations. Simple user-friendly games, such as matching, Tic-tac-toe, Jeopardy, and engineering-based simulations are created to keep users active, excited, and interested in the learning process. Special engineering and technical simulations are developed based on the instructional needs to deliver the concept effectively, and guide the user through the investigation process. Another advantage of using simulations is that users can try different options without the risk of damaging costly equipments and components or getting some unexpected results. These activities are monitored to provide instant feedback throughout the learning process and to instruct the learners' behaviors. Simulations are in both two and three dimensions as shown in Figures 6.19a and 6.19b.


OC-ALC Online Training System
C-135 Cabin Pressurization Familiarization
Equipment location exercise
Manual control valve
Cabin pressure controller
Manual control valve
Pressure test valve
Outflow safety valve
Pneumatic relay
Outflow safety valve
Pneumatic relay
Frame 1 2 3 4 5 6 7 8 9 10 11
Objective 1 2

The pneumatic relay is located here.


OC-ALC Online Training System
Tools and Equipment - 10.9 Proof Test Procedures
10.9 Proof Test virtual scene
Zoom In
Zoom Out
Reset
Frame 1 2 3 4 5 6 7 8
Objective 1 2

Click on a model to see what each represents

Safety Cones placed to create a 25ft perimeter on all sides from aircraft. Once the actual pressurization procedure begins, no one other than team personnel are allowed inside the perimeter (10.9).

Figures 6.19a and 6.19b: Examples of simulation

Two-dimensional simulations are only integrated in the EM/C135 System as primary learning technology for users. Two-dimensional simulations are employed to give warnings and explanation of rules and procedures in the C135 aircraft maintenance and environment management. Some of the two-dimensional simulations are even designed as games such as hang man, matching game, Tic-tac-toe and Jeopardy. These simulations improve the learners' interest and engage the students with the activity. Regardless of whether the simulation is designed as a game or as normal theory explanation, the goal of simulation is to clarify a technical concept.

One of the unique features in the demonstration systems that are implemented in this research is high-level three-dimensional simulation. Although three-dimensional simulations are integrated in the EM/C135 System as well, those simulations are only basic-level three-dimensional simulations. In other words, three-dimensional simulations in the EM/C135 System are reasonably straightforward but three-dimensional simulations in the demonstration systems are extremely complicated in terms of implementation and function as well. Compared with the two-dimensional simulation, the three-dimensional simulation better mimics the real world and is able to explain concepts more clearly. All simulations both in two and three dimensions are developed using *Director* and are fully controlled by built-in Lingo scripting language.

2.5 Sound

Sound is important to enhance the learning opportunities of users. Sound is used in the system for several purposes; first as a narration, and then to give feedback to users' response in simulations. Sound narration is used in every frame to underline ideas and concepts presented as text or graphics. The changing of tones in the narration emphasizes the important ideas in the frame and catches the attention of the learners. Further, the sound summarizes and explains the textual content in each frame. It is used to give complete instructions of a simulation or a brief summary of evaluation questions. The lengths of the audio clips are neither too long nor too short, so that the user's attention is not diverted from the main content. The sound intensity for the feedback effects is maintained below the maximum level of the narrations, so that no shock is caused due to a sudden alert. The audio is recorded in *wav* format, and then converted to *swa* format by *Director*, later it is linked and streamed by *Director* to synchronize with other media elements.

2.6 Video

Video is used to illustrate examples from the actual working environment to give a practical approach towards training. Video is integrated in the course because it can explain complex procedures with real and accurate graphics. Although video is characterized by large file size, it is used to illustrate examples from the actual working environment to give a pragmatic approach towards training. In order to maintain the quality to support a smooth web delivery, the movie clips are compressed and maintained to short durations of 10-30 seconds. Figure 6.20 shows an example of this technology.

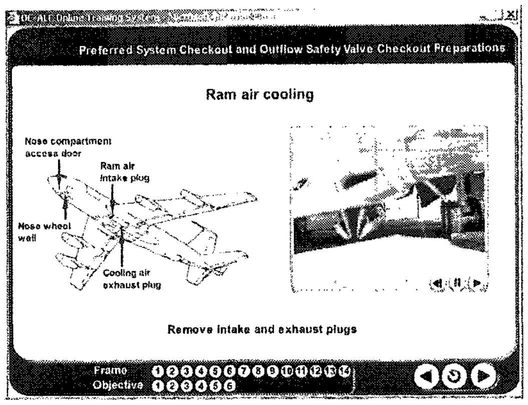

Preferred System Checkout and Outflow Safety Valve Checkout Preparations
Ram air cooling
Nose compartment access door
Ram air intake plug
Nose wheel well
Cooling air exhaust plug
Remove intake and exhaust plugs
Frame
Objective

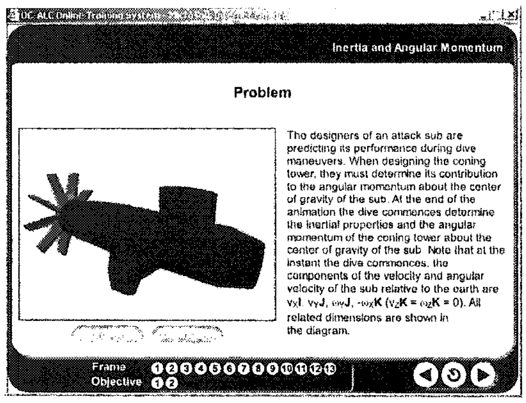
Figure 6.20: Incorporation of Video Element

In order to make it convenient for learners to watch and review the video, play, stop, rewind, and slider buttons were created. The original video clip is obtained from previous work recorded by Tinker AFB with a digital video camera. The video was imported from the camera to a computer and stored in *QuickTime* format. Before movies are integrated into the *Director* file, *QuickTime* movie has to be converted into a *Flash* movie in order to solve the problem caused by Tinker firewall that blocks *QuickTime Xtra*, which is necessary for *QuickTime* playing inside *Director*.

2.7 Three-Dimensional Visualization

Three-Dimensional Visualization is used to help users better understand C-135 aircraft maintenance and instruction. This is only a preface of three-dimensional visualization that is integrated in the EM/C135 System. An advanced level of three-dimensional visualization is heavily incorporated in the learning environment throughout the demonstration systems as the unique feature of this research with the introduction of

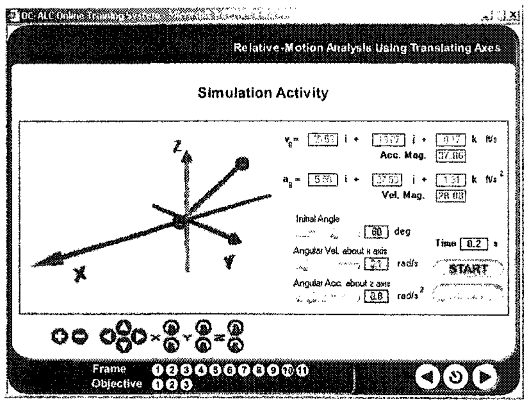
Shockwave 3D technology. This allows for the development of a wide spectrum of three-dimensional productions, ranging from simple text handling to interactive product demonstrations to complete complex game environments, which include animation and simulation. Figures 6.21a and 6.21b show examples of three-dimensional animation and simulation respectively.



DC-ALC Online Training System
Inertia and Angular Momentum
Frame 1 2 3 4 5 6 7 8 9 10 11 12 13
Objective 1 2

Problem

The designers of an attack sub are predicting its performance during dive maneuvers. When designing the coning tower, they must determine its contribution to the angular momentum about the center of gravity of the sub. At the end of the animation the dive commences determine the inertial properties and the angular momentum of the coning tower about the center of gravity of the sub Note that at the instant the dive commences, the components of the velocity and angular velocity of the sub relative to the earth are $v_X\mathbf{I}$, $v_Y\mathbf{J}$, $\omega_Y\mathbf{J}$, $-\omega_X\mathbf{K}$ ($v_Z\mathbf{K} = \omega_Z\mathbf{K} = 0$). All related dimensions are shown in the diagram.



DC-ALC Online Training System
Relative-Motion Analysis Using Translating Axes
Z
X
Y
START
Frame 1 2 3 4 5 6 7 8 9 10 11
Objective 1 2 3

Simulation Activity

v_B = [illegible] i + [illegible] j + [illegible] k ft/s

Acc. Mag. 37.86

a_B = 5.50 i + 37.53 i + 1.31 k ft/s^2

Vel. Mag. 28.03

Initial Angle 60 deg

Angular Vel. about x axis [illegible] rad/s

Angular Acc. about z axis 0.8 rad/s^2

Time 0.2 s

Figures 6.21a and 6.21b: Example of three-dimensional animation and simulation

CHAPTER 7: SUMMARY AND CONCLUSIONS

The primary objective of this research is to investigate, demonstrate, and evaluate the effectiveness of implementing three-dimensional visualization in an interactive web-based environment for engineering education and technical training using the latest three-dimensional supportive capability in the Shockwave 3D technology. The other potential objectives in conducting this research is to provide a foundation for developing content rich-material for engineering education and technical training, and an evaluation of the efficiency of the application in delivering engineering and technical concepts through the Internet. To achieve these research objectives, an illustration environment needs to be designed, developed, implemented, and assessed. This demonstration environment is mainly incorporated using three-dimensional visualization collaborating with other multimedia technologies to present an interactive web-based environment for engineering education and technical training. These various technologies, which are employed to design and develop the environment, together formed the basis of the environment and are integrated to communicate smoothly with each other to present the learning materials clearly. Among these technologies, the Shockwave 3D technology is considered the main and unique feature implemented.

Shockwave 3D is a technology used to develop applications with complicated static and dynamic three-dimensional graphics. This is a technology that can eliminate the obstacles faced using other existing technologies such as VRML and Java 3D, which are the current technologies mainly used in several-presented web-based environments. The advantages of employing the latest technology with the web-based three-dimensional graphics supportive capability, namely Shockwave 3D, are its inbuilt streaming capabilities, effortlessness in finding an appropriate widely installed plug-in, and simplification of implementing and understanding this technology. Shockwave 3D is the extended feature of the extensively used Shockwave technology, which is developed using the latest version of Director software, widely known as Director 3D. The advantages of using Shockwave 3D that has been investigated in this research, compared to other technologies of its kind are the main factors that this technology is introduced as major technique used to develop an effective web-based environment.

The content of this demonstration environment is carefully determined to ensure that three-dimensional visualization, which is the most important element required reaching the objectives, are integrated successfully. Two demonstration environments are developed separately for technical training and engineering education purposes. The content of the technical training environment concentrates on the topic of *Aircraft Part Maintenance and Assembly Process*; this environment is also part of the *CSD Part Assembly Training Demo Using Interactive CAD* project funded by Tinker Air Force Base (AFB). A primary objective of this project is to develop an interactive web-based environment that primarily integrates with three-dimensional animations and simulations to facilitate maintenance activities. This project, which is the development of an immersive and interactive virtual prototyping environment and process for maintenance, includes the delivery of two main components. First, an interactive web-based application that provides three-dimensional maintenance and training instructions. Second, a demonstration of developed system applicability uses the *Constant Speed Mechanical Drive (CSMD) Mechanism* part assembly. This environment is identified as the CSD System throughout the research. The engineering education environment, namely the 3D

Dynamics System, focuses on the topic of *Three-dimensional Rigid Body Motion*, which is one of the major topics in the *Engineering Dynamics* course required in most undergraduate engineering curriculums.

1. Demonstration Environments

1.1 CSD System

The interactive web-based training system is adopted because Tinker AFB has realized how convenient, flexible, effective, easily accessible, and low-cost this system is in concurrently training its personnel, while it is still capable of supporting a participation intensive learning style identical to traditional training methods. The demonstration is the actual system used to train and guide Tinker AFB personnel to properly perform procedures for the specific *Aircraft Part Maintenance and Assembly Process*. Once training is completed, the user has the ability to accurately operate the specific *Aircraft Part Maintenance and Assembly Process* as a result expected by Tinker AFB. Even though this system is the actual system developed for Tinker AFB, it is only at a demonstration stage to explore the possibility of launching a complete interactive web-based training system in the future. This demonstration environment focuses only on the *Left Hand Pump and Motor (LHPM) Assembly*, which is a subassembly of the *CSMD Mechanism*. The system consists of three major sections: the *Part Breakdown* section, the *Assembly Process Tutorial* section, and the *Assembly Process Evaluation* section that can be compared to the evaluation of a traditional training method.

The environment is composed of three major elements, including interactivity, three-dimensional visualization, and evaluation. These elements are considered essential elements for a web-based environment, and they are the minimum required elements needed for this research to achieve the objectives and to comply with Tinker AFB requirements as well. Interactivity is maintained throughout the learning process to make the system participative intensive and so that the concentration of the user is focused on the subject. This element is mainly represented by a variety of simulations that are designed and implemented to enable students to learn through exploration and investigation along with the learning material. Three-dimensional visualization is integrated to develop and deliver content rich learning material efficiently. This element is used as the main component universally in the system, which means it is also the principal piece of work used to build the other two essential elements, including interactivity and evaluation.

Evaluation, which developed based on the other two essential elements, interactivity and three-dimensional visualization, is represented through various high level interactive simulations. These high level interactive simulations are designed to evaluate the effectiveness of the user's learning capability by allowing the user to perform the maintenance process as presented in the *Assembly Process Tutorial* section or animation section and to provide feedback to the user. This evaluation procedure is located at the end of a learning block, which corresponds to an entire assembly process for a single assembly part. Each step inside the *Assembly Process Tutorial* section can be compared relatively to an official Technical Order (TO) provided by Tinker AFB. Finally, considering the system in an implementation aspect, this entire system is developed inside Shockwave 3D and is implemented using only Lingo scripting language to control simulation, animation, three-dimensional model movements, navigation system, and other interactivities that appear during the running of the system.

Key expected benefits and impacts by Tinker AFB from using this system are:

- The system can provide interactive training or retraining of personnel for maintenance activities.
- The system can provide users the opportunity to interact with subsystems and subcomponents to give a better understanding of the process.
- The system can provide Tinker AFB personnel anytime, anywhere access to maintenance instructions, enhanced through simulations, three-dimensional CAD models, and other relevant information.
- The system can provide access to all related information for maintenance processes, such as digital CAD models data, process information, manufacturing data, etc.
- The system can provide animations, simulations, and information that enhance understanding of the maintenance process by personnel.
- The system can reduce the time needed to train Tinker AFB personnel through direct interaction with processes and digital examples of maintenance procedures that need to be maintained.
- The system can provide instructions to personnel in order to refresh memory.
- The system can provide a flexible workforce at Tinker AFB that can be quickly retrained and reorganized to reduce time for maintenance activities.

1.2 3D Dynamics System

The 3D Dynamics System is an interactive web-based educational environment developed to assist the student in learning basic concepts in *Engineering Dynamics*. The environment concentrates only on *Three-dimensional Rigid Body Motion*, and covers the same material addressed in a typical, undergraduate *Engineering Dynamics* book and the course website (www.ecourses.ou.edu); it is designed to serve as a third source in addition to lectures and textbooks from which engineering students can learn about *Engineering Dynamics.* The system is considered an extended topic of the EM/C135 System, and is composed of two lessons illustrating specific concepts in *Three-dimensional Rigid Body Motion*: *Three-dimensional Kinematics of a Rigid Body* and *Three-dimensional Kinetics of a Rigid Body*. Each lesson consists of a number of objectives to cover all the major topics in a fashion that is similar to that of a typical *Engineering Dynamics* textbook. Moreover, each particular section is divided into three areas, including the *Theory* section, the *Simulation Activity* section, and the *Problem* section in order to provide students with a linear learning process for clearly understanding concepts.

With the 3D Dynamics System, the focus has been to integrate computer-based simulations with graphics, audio components, animations, and hypertext, into an interactive, self-contained system that is accessible through the Internet to assist students taking an introduction *Dynamics* course. The goal is to embody the theoretical and discursive aspects of *Dynamics* with experimental simulations in the same system. The result for students will hopefully be a more efficient learning process and a deeper understanding of *Dynamics*. Also, this system can hopefully provide students with a more efficient means of understanding *Engineering Dynamics.* One advantage of computer-based simulations is that they allow students to experiment with a larger range of problems and situations, and they provide more immediate feedback. In addition, computer-based simulations are generally far more economical than laboratory or shop experience. However, hands-on experience is still invaluable for a comprehensive education. Ideally, students should be exposed to a combination of both mechanical and computer-based simulations.

1.3 Accomplishments

This research has led to the development of an interactive web-based environment for engineering education and technical training. This research investigates the use of the latest technology that has advantages over any existing technologies for the developer to develop and for the user to experience more effectively and conveniently a web-based environment in order to better understand the technical concepts clearly through the presentation of illustration environments. This is the first interactive web-based environment concentrating on engineering education and technical training that mainly involves the performance of three-dimensional visualization using the latest three-dimensional supportive capability technology in Shockwave 3D. An implementation of three-dimensional visualization using the Shockwave 3D technology is emphasized in the development of a complex application with the integration of numerous Internet and multimedia technologies that work with each other smoothly in order to attain the goals of this research. In addition, it is the first framework for engineering education and technical training in the topics of *Aircraft Part Maintenance and Assembly Process* and *Engineering Dynamics: Three-dimensional Rigid Body Motion* that employs the capability of the Shockwave 3D technology. This demonstration environment is developed because both engineering education and technical training need prototype applications that utilize the Internet and web technology, along with the implementation of various multimedia and engineering design technologies, in order to potentially improve the delivery of learning materials.

1.4 Contributions

The contributions of this research mainly cover the idea of utilizing the advantage of three-dimensional visualization in explaining technical and engineering technologies and concepts clearly. The environment is an educational system with rich content that can be used to teach multiple engineering and technical topics with the assistance of three-dimensional visualization including graphics, animation, and simulation. A hierarchical organization structure is designed to ease the system management and flexible enough for modification and extension of the content in order to teach various specific engineering and technical topics. The interface is user-friendly and the environment is straightforward with intuitive illustration that gives a direct impression to users in demonstrating engineering and technical concepts.

The concept of using the latest technology as a unique feature to implement three-dimensional visualization for more effective and convenient web-based environment delivery is investigated and implemented. Shockwave 3D technology is the latest technology with web-based three-dimensional graphics supportive capability, as a primary tool to create, develop, and implement a progress performance for an interactive web-based training environment. The key features of employing this technology are its inbuilt streaming capabilities, effortlessness in finding an appropriate and widely installed plug-in, its ability to be accessed through the web easily, its simple development process and simplification of implementing and understanding this technology. These are the main factors to make this technology a more appropriate tool in developing three-dimensional visualization for the web-based environment. In addition, this research clearly and productively expands Shockwave 3D's capabilities to the world of education and training on the web. With its abilities, this technology can bring web-based education and training to the next level of effectiveness. Also, unique programming techniques are employed in the development of the environment. This significantly increases the readability of the codes.

1.5 Assessment

Assessment is conducted over a period of the development process for both demonstration environments. For the CSD System, in order to assess the effectiveness of the system, the progress of the project has been reported and presented to Tinker AFB's project coordinators on a frequent basis in order to comply with the requirements of Tinker AFB. Reviews, comments, feedback, and suggestions from Tinker AFB on any related features, such as the training contents, system operation, system features, and organization, are received and reviewed after each section is developed.

Moreover, the system has been demonstrated to actual users and their supervisors a number of times as well. Several issues were involved in the presentation, including how to use the system, what the system is for, what users will experience from the system, and why the system is significant to the Tinker AFB personnel who are responsible for aircraft part maintenance and the assembly process. The demonstration focused on a group of users who have to work on the *CSMD Mechanism*. However, the demonstration was not limited to this group of people; it was also presented to other groups of people who have to work on other parts of the aircraft. The only thing that is different is the content inside the system; the rest is the same as the CSD System. Other groups of people who are not involved with aircraft part assembly and maintenance also attended some of the demonstrations because they want to know how an interactive web-based training system will benefit their departments, such as the information technology department.

The feedbacks after the demonstrations were extremely positive as to the way the system can contribute to employees' work and responsibility. The main group, or the *CSMD Mechanism* shop floor persons who initiated this project, were pleased with the way the system works and they decided to continue the project to the second phase. The second phase concerns simplifying the CSD System development process in order to more easily develop a similar system for other parts. This project is in the beginning process of investigation at the time of this writing. However, all the positive feedbacks received from Tinker AFB personnel are verbal only; there was no survey or mathematical statistics involved. However, since Tinker AFB decided to continue the project to the next level, clearly the CSD System has impressed Tinker AFB in its performance and effectiveness, as was expected.

This process assists in developing the system in order to conform to the requirements specified by Tinker AFB. In addition, a complete demonstration environment will be evaluated in the assessment as well after Tinker AFB takes delivery of the system. The assessment will be conducted by both the actual group of Tinker AFB personnel who are assigned to utilize this system and also the group of project development members, or project coordinators, at Tinker AFB in order to evaluate learning effectiveness. The results of these tests will be collected and analyzed statistically. Currently, this system is hosted at the University of Oklahoma, but the complete system will be hosted inside the base for use by air force personnel only. The option to allow the public to access the system is left to Tinker's consideration. The change of location of the learning system can be done easily by physically installing the system in the base server, after which Tinker AFB will test real users as part of the evaluation.

The 3D Dynamics System will be evaluated by both groups of students who enroll in classes involved with this particular topic and professors who teach this specific subject at the University of Oklahoma when this class is offered in the Spring of 2004. The results of this evaluation will be collected and analyzed statistically and will presented in the future as well. However, during development, the system went through testing for both functionality and proof of concept.

The proof of concept was tested several times, including areas such as the usability of the interactive and the usefulness of the system when integrated into a curriculum. Numerous suggestions were provided to improve the system for students' learning. By the last testing, the results were positive, indicating the effectiveness of the system. The system has a high tendency to provide students with an enjoyable better understanding of the physical concepts of *Dynamics*. However, to get a more complete analysis of the usefulness of the system when integrated into a curriculum, it is necessary to perform controlled experiments in a classroom setting with actual students and professors.

The test of using the system will be in an actual classroom setting. During the 2004 Spring semester at the University of Oklahoma, the system will be integrated into one of the basic *Dynamics* course required of all engineering students. The course will be taught by the author's advisor who will introduce the system as additional learning tool for students. The system is not designed to replace the course's textbook or lectures, but instead to provide an extra learning resource. Students will be told to use the system outside of the classroom. The system is not designed to be a lecture tool and thus will not be used during lectures. To encourage students to use the system, students will be told that test questions could come from the system. At the time of this writing, the semester has not yet begun, thus the final results as to whether the system will actually contribute to educating students more efficiently are not known at this time. The final results of the classroom testing will be presented at the conference, which will include a survey and the test results. Moreover, the author plans to compare students' test scores with other *Dynamics* courses taught by the same instructor.

1.6 Recommendations for Future Work

The CSD System and the 3D Dynamics System are demonstrations to show how technical engineering subject can be explained through the interactive web-based environment effectively since the three-dimensional visualization is completely employed as a major element. There are several research topics that can improve the performance of this training system because these systems are still at the demonstration phase. However, its concepts can provide the basis of a virtual university for engineering education and technical training. In order to improve the performance of these systems, additional features are recommended such as users' ability to create simple three-dimensional objects for their own experiment in the 3D Dynamics System and the function that can combine various subassemblies presented in different modules into an entire assembly for the CSD System. Another issue for the improvement of the CSD System is the creation of three-dimensional models using the Lingo Scripting capability available in Shockwave 3D technology. Because at this time, three-dimensional models used in the system were created using another application and then imported to the Director in order to be used by Shockwave 3D technology. This results in the big file size of three-dimensional models and that requires a period of time for the system to download all the models. This is a significant concern in terms of downloading time, file size, and memory usage. Using Lingo Scripting to create three-dimensional models would help solve this problem; however, creating such complex three-dimensional models is a very challenging task since the structure of the Lingo Scripting does not provide sufficient flexibility and convenience for the developer. The developer has to clearly understand the mathematical representation of three-dimensional models.

Currently, the CSD System and the 3D Dynamics System mainly focus on two different topics. Therefore, future research may extend to topics in other engineering and technical disciplines based on the framework and development of these systems. Additional carefully designed evaluation strategies are needed to assess the effectiveness of the entire system. Control groups may be used in the assessment.

2. EM/C135 System

2.1 Assessment

In order to assess the effectiveness of the system, the progress of the project has been reported and presented to Tinker's project coordinators on a frequent basis in order to comply with the requirements of Tinker. Reviews, comments, feedback, and suggestions from Tinker on any related features, such as the lesson contents, system operation, system features, and organization, were received and reviewed after each module was developed. This process assisted in developing the system to conform to the requirements specified by Tinker. Currently, the complete system is hosted inside the base for use by air force personnel only. The option to allow the public to access the system is left to Tinker's consideration. The transferring of the system was done easily by physically shifting the server formerly used as the host server from the University of Oklahoma to the base. Changing the host name and IP address for the server and connecting a database to the system completed the installation process of the web-based training system. Refer to Tinker by the time of system transfer, there will be a test conducted on actual users to evaluate the system's effectiveness. The complete results of this test will be kept secret on the base but the summary of the test will be reported to the development team members at the University of Oklahoma.

2.2 Conclusion

The web-based training system for Environmental Engineering and C-135 Aircraft Maintenance Instruction was developed for Tinker AFB to support participative, intensive learning. The objective of the Environmental Engineering modules is to train and guide Tinker AFB personnel to comply with the standard requirements of various NESHAP (National Emission Standard for Hazardous Air Pollutants) and environmental regulations. Meanwhile, the objective of the C-135 Aircraft Maintenance Instruction is to train and guide Tinker AFB personnel to correctly perform the procedures required in maintaining pressurization on the C-135 series aircraft. The major differences between these two topics are the lesson content and the fact that aircraft maintenance training incorporates three-dimensional visualization features.

The system is comprised of three main segments: the lessons, the administrative system, and the database. The course content is separated into six modules or topics. The modules are then subdivided into lessons that contain objectives that focus on specific training material. Each lesson integrates text, graphics, video, animations, simulations, and sound to deliver the course content. At the end of each objective, users are asked to answer questions related to the material reviewed and the results are recorded in the database.

A web compatible database was designed and integrated into the training system to support online information tracking and management. The delivery of the lessons and tracking of the user progress are controlled using a database. All information is stored within the database and is critical to the deliver of the lessons. The database is maintained and edited through the administrative web pages. A hierarchical organization structure was designed to ease the system management. An administrative section allows administrators and managers to set up user accounts, manage lessons, create user groups, and report on user progress. All administrative programs are web-server based and can be operated through any web browser.

REFERENCES

[1] John S. Mccright, PC Week Online, “Cisco’s Chambers: e-learning will help us control our destinies”, http://www.zdnet.com/pcweek/, Nov. 16, 1999

[2] Debra Donstron, “From the Trenches: Distributed Learning Is High Priority”, PC Week, November 14, 1999

[3] Qiuli Sun and Kurt Gramoll, “Internet-based Distributed Collaborative Environment for Engineering Education and Design”, 2001 ASEE Annual Conference & Exposition, Albuquerque, New Mexico, June 24 - 27, 2001

[4] Karthik Ranga and Kurt Gramoll, “Design Education over the Internet using VRML”, 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[5] Karthik Ranga and Kurt Gramoll, “3-D Finite Element Analysis on the Internet using Java and VRML,” 2000 ASEE Annual Conference & Exposition, St Louis, Missouri, June 18-21, 2000

[6] Mohammed E. Haque, “Web-based Visualization Techniques for Structural Design Education”, 2001 ASEE Annual Conference & Exposition, Albuquerque, New Mexico, June 24 - 27, 2001

[7] Kian-Huat Tan, Tze-Leong Yew, and Kurt Gramoll, “Understanding Machine Operations and Manufacturing using VRML”, 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[8] Qiuli Sun, “Internet-based Distributed Collaborative Environment for Engineering Education and Design”, PhD Dissertation, The University of Oklahoma, 2001

[9] Martin McCarthy and Alligator Descartes, "Reality Architecture: Building 3D worlds with Java and VRML", Prentice Hall Europe, Hertfordshire, United Kingdom 1998

[10] Rodger Lea, Kouichi Matsuda, and Ken Miyashita, "Java for 3D and VRML Worlds", New Riders Publishing, Indianapolis, Indiana 1996

[11] D J. McArthur, and M. W. Lewis, "Untangling the Web: Applications of the Internet and Other Information technologies to Higher Learning", pp.1-20, 1998

[12] Hillman, A. L. and Wells, D.M, "Breaking Down the Barriers: Using Web-Based Instruction at the University Level--Mississippi State University"

[13] Yacob Astatke, "Creating a Distributed Learning Environment using WebCT", 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[14] William J. Strenth, "Developing "Civil Construction", An Internet Class using Electronic Blackboard", 2001 ASEE Annual Conference & Exposition, Albuquerque, New Mexico, June 24-27, 2001

[15] Arun Arunachalam, "Design and Development of a Web Enabled Interactive System to Support Environmental Engineering Training", MS Thesis, The University of Oklahoma, 2001

[16] Charles R. Graham and Timothy N. Trick, "Java Applets Enhance Learning in a Freshman ECE Course", Journal of Engineering Education, Vol.87, No. 4, October 1998, pp.391-397

[17] W.J. Austin, J. Liddle, R.C. Thomas, and P. McAndrew, "Networked Educational Simulation in Java", 1998 International Conference on Web-Based Modeling & Simulation, Simulation Series Vol. 30, No. 1, 1998, pp.105-109

[18] R.C. Thomas, J. Liddle, W.J. Austin, and P. McAndrew, "Multiverse-Lowering the Barriers to Widespread Use of Web-based Simulations in Education", 1998 International Conference on Web-Based Modeling & Simulation, Simulation Series Vol. 30, No. 1, 1998, pp.110-113

[19] Robert Desharnais and Gary Novak, "Virtual Courseware for Science Education", Syllabus, Vol. 12, No.1, August 1998

[20] Qiuli Sun, Kurt Gramoll, and Michael Mooney, "Self-Paced Instruction to Introduce Traffic Engineering in Virtual City (Sooner City)", 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[21] Qiuli Sun and Kurt Gramoll, "Internet-based Simulation and Virtual City for Engineering Education", 2000 ASEE Annual Conference & Exposition, St Louis, Missouri, June 18-21, 2000

[22] A. J. Baker, Z. Chambers, and M. B. Taylor, "Finite Element Analysis for the Engineering Sciences: A Web-based, Video-streamed Education Environment at a Distance", 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[23] Terry H. Walker, Z. Chambers, M. B. Taylor, and A. J. Baker, "Finite Element Analysis for Biological Engineering: A Web-based, Distant Education Venue", 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[24] Monson H. Hayes and Lonnie Harvel, "Distance Learning into the 21st Century", 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[25] Kurt Gramoll, "Teaching Statics Online with only Electronic Media on Laptop Computers", 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[26] Yellamraju Vikas, Tony Romanello, and Kurt Gramoll, "Teaching Dynamics Online with only Electronic Media on Laptop Computers," 2000 ASEE Annual Conference & Exposition, St Louis, Missouri, June 18-21, 2000

[27] Adrian Ng and Kurt Gramoll, "Online Review and Practice Tests for the Fundamentals of Engineering Exam", 1999 ASEE Annual Conference & Exposition, Charlotte, North Carolina, June 20-23, 1999

[28] Steen Peterson, "Using Java and VRML in Teaching Machine Kinematics", Ninth Annual Conference – Technology-based Engineering Education Consortium, Nov. 21-22, 1997, Vanderbilt University, Nashville, Tennessee

[29] http://www.explorelearning.com/index.cfm

[30] Rujin Cheng, "Web-based Distance Learning Environment for Computer Aided Engineering Design and Analysis Tools", MS Thesis, The University of Oklahoma, 2002

[31] David R. Wallace and Philip Mutton, "A Comparative Evaluation of World Wide Web-Based and Classroom Teaching", Journal of Engineering Education, Vol.86, No. 3, July 1998, pp.211-219

[32] David Wallace and Suzanne Weiner, "How might classroom be used given WWW-based lectures? A comparison of a lecture-style second coverage of materials vs. limited-coverage guided experiential activity", ASEE Journal of Engineering Education, volume 87, number 3, 1998, pp. 237-248

[33] Rosemarie M. Evans, Madison Daily, and Susan L. Murry, "The Effectiveness of an Online Graduate Engineering Management Course: A Preliminary Study", 1998 ASEE Annual Conference & Exposition, Seattle, Washington, June 28-July 1, 1998

[34] William E. Cole, "Using CAD Analysis Tools to Teach mechanical Engineering Technology", 1998 ASEE Annual Conference Proceedings, 1998

[35] John T.Bell and H. Scott Fogler, "Ten Steps to Developing Virtual Reality Applications for Engineering Education", 1997 ASEE Annual Conference Proceedings, 1997

[36] Chaturaporn Nisagornsen, Arun Arunachalam, Kurt Gramoll, and Hengzhong Wen, "Interactive Web-Based Training System at Tinker AFB: Environmental Engineering, and C-135 Aircraft Maintenance Instruction", 2002 ASEE Annual Conference & Exposition, Montréal, Quebec Canada, June 16-19, 2002

[37] Hengzhong Wen, "Development and Implementation of Web Based Training for Tinker Air Force Base", MS Thesis, The University of Oklahoma, 2002

[38] Tinker Air Force Base, "Technical Manual – Overhaul Instructions: Constant Speed Mechanical Drive", Midwest City, Oklahoma 2002

[39] Tinker Air Force Base, "Technical Manual – Illustrated Parts Breakdown: Constant Speed Mechanical Drive", Midwest City, Oklahoma 1999

[40] R.C. Hibbeler, "Engineering Mechanics Dynamics", 9th edition, Prentice-Hall, Inc., Upper Saddle River, New Jersey 2001

[41] J.L. Meriam and L.G. Kraige, "Engineering Mechanics Dynamics", 5th edition, John Wiley & Sons, Inc., New York, New York 2002

[42] Kurt Gramoll and Rob Abbanat, "Interactive Multimedia for Engineering Dynamics", 1995 ASEE Annual Conference & Exposition, Anaheim, California, June 25-28, 1995

[43] http://emet.ou.edu/engin/home.htm

[44] Dean Utian, Sam Humphries, Charles Parcell, Jose R. Rodriguez, Chuck Wainman, II, and Luke Wigley, "Foundation Director 8.5", friends of ED, Chicago, Illinois 2001

[45] Gary Rosenzweig, "Special Edition Using Director 8.5", Que, Indianapolis, Indiana 2002

[46] Phil Gross and Mike Gross, "Macromedia Director 8.5 Shockwave Studio for 3D: Training from the Source", Macromedia Press, Berkeley, California 2002

[47] Jay Armstrong, Barbara Herbert, and Stephanie Gowin, "Macromedia Director 8.5 Shockwave Studio Using Director 8.5 Shockwave Studio", 2nd edition, Macromedia, Inc., San Francisco, California 2001

[48] Jay Armstrong, Barbara Herbert, Stephanie Gowin, Tom Higgins, Macelle Taylor, and Frank Welsch "Macromedia Director 8.5 Shockwave Studio What's New in Director Shockwave Studio", 1st edition, Macromedia, Inc., San Francisco, California 2001

[49] http://www.macromedia.com/software/dreamweaver/

[50] http://www.macromedia.com/software/director/

[51] http://www.macromedia.com/software/director/3d

[52] http://www.macromedia.com/software/flash/

[53] http://www.adobe.com/products/photoshop/main.html

[54] http://mediasoftware.sonypictures.com/products/soundforgefamily.asp

[55] http://www.microsoft.com/sql/default.asp

[56] About.com, "SQL fundamentals".
http://databases.about.com/library/weekly/aa020401a.htm?PM=ss14_databases

[57] McFadden, F., J. Hoffer, and M. Prescott. "Modern Database Management", Addison-Wesley, Berkeley, California 1999

[58] http://www.asp101.com/

[59] http://www.aspin.com/

[60] Microsoft Corporation, "VBScript user's guide".
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vbstutor.asp

[61] David Flangan, "JavaScript – The Definite Guide", O'Reily & Associate, Inc., Sebastopol, California 1997

[62] Nathan Segal, (July 9, 2003), "Swift 3D: Vector Animation for the Web".
http://www.webreference.com/3d/column2/

[63] Jesse Nieminen, "Product Review: Electric Rain Swift 3D 2.0".
http://www.digital-web.com/reviews/product/review_2002-05.shtml

[64] Lee Creek, (February 6, 2002), "Swift 3D".
http://wdvl.internet.com/Reviews/Graphics/Swift3D/

[65] Jim X. Chen, "Guide to Graphics Software Tools", Springer-Verlag New York, Inc., New York, New York 2003

[66] http://www.discreet.com/products/3dsmax/

[67] http://www.erain.com

[68] http://www.newtek.com/

APPENDIX A: LINGO SCRIPT FOR ANIMATION

```
-- The "finalizeStep" function defines which part has to be
-- animated, the direction of the animation, and the final part position
-- for each step in the animation section. This function also calls
-- the "animateSubAssembly" function as a sub-routine function to
-- complete each individual part animation.
on finalizeStep stepNo

  -- ** ANIMATE OBJECT **
  animatedVec = vector(0, 0, 0.05)
  finalPos    = vector(0, 0, 0)

  if stepNo = 1 then
    animateSubAssembly(obj706620, -animatedVec, finalPos) -- ** M-10 **
  else if stepNo = 2 then
    -- ** NO ANIMATION **
  else if stepNo = 3 then
    animateSubAssembly(obj706606, animatedVec, finalPos) -- ** T-24 **
    if obj706606.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj712113, animatedVec, finalPos) -- ** T-23 **
      sprite(spriteList.getAt(14)).loc = originalPos
      member("706606 Text").color      = blackColor
      sprite(spriteList.getAt(21)).loc = changedPos
      member("712113 Text").color      = changedColor
    end if
    if obj712113.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj706610, animatedVec, finalPos) -- ** T-22 **
      sprite(spriteList.getAt(21)).loc = originalPos
      member("712113 Text").color      = blackColor
      sprite(spriteList.getAt(15)).loc = changedPos
      member("706610 Text").color      = changedColor
    end if
    if obj706610.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj3415_90P, animatedVec, finalPos) -- ** T-21 **
      sprite(spriteList.getAt(15)).loc = originalPos
      member("706610 Text").color      = blackColor
      sprite(spriteList.getAt(8)).loc  = changedPos
      member("3415-90P Text").color    = changedColor
    end if
  else if stepNo = 4 then
    -- ** NO ANIMATION **
  else if stepNo = 5 then
    animatedVec = vector(0.05, 0, 0)
    animateSubAssembly(obj711430, animatedVec, finalPos) -- ** B-42 **
  else if stepNo = 6 then
    -- ** OPTIONAL PARTS **
    animateSubAssembly(obj710641, animatedVec, finalPos) -- ** B-37 **
    animateSubAssembly(obj710641_02, animatedVec, finalPos)
    if obj710641.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj713737, animatedVec, finalPos) -- ** B-38 **
      animateSubAssembly(obj713737_02, animatedVec, finalPos)
      sprite(spriteList.getAt(19)).loc = originalPos
      member("710641 Text").color      = blackColor
      sprite(spriteList.getAt(27)).loc = changedPos
      member("713737 Text").color      = changedColor
    end if
    if obj713737.transform.position = vector(0, 0, 0) then
      animateSubAssembly(objMS21043_06, -animatedVec, finalPos) -- ** B-39 **
      animateSubAssembly(objMS21043_06_02, -animatedVec, finalPos)
      sprite(spriteList.getAt(27)).loc = originalPos
      member("713737 Text").color      = blackColor
      sprite(spriteList.getAt(38)).loc = changedPos
      member("MS21043-06 Text").color  = changedColor
    end if
  else if stepNo = 7 then
    -- ** OPTIONAL PARTS **
    animateSubAssembly(obj711430, animatedVec, finalPos)
    animateSubAssembly(obj739497, animatedVec, finalPos)
    animateSubAssembly(obj710641, animatedVec, finalPos)
    animateSubAssembly(obj710641_02, animatedVec, finalPos)
    animateSubAssembly(obj713737, animatedVec, finalPos)
    animateSubAssembly(obj713737_02, animatedVec, finalPos)
    animateSubAssembly(objMS21043_06, animatedVec, finalPos)
    animateSubAssembly(objMS21043_06_02, animatedVec, finalPos)
    if obj711430.transform.position = vector(0, 0, 0) then
      aVec  = vector(0.05, 0, 0)
      fPos  = vector(xDistance * 1, 0, zDistance * 0.5)
      fPos2 = vector(-xDistance * 1, 0, zDistance * 0.5)
      animateSubAssembly(obj4004C4_6_14, -aVec, fPos)
      animateSubAssembly(obj4004C4_6_14_02, aVec, fPos)
      sprite(spriteList.getAt(31)).loc = originalPos
      member("739497 Text").color      = blackColor
      sprite(spriteList.getAt(11)).loc = changedPos
      member("4004C4-6-14 Text").color = changedColor
    end if
    if obj4004C4_6_14.transform.position = vector(xDistance * 1, 0, zDistance * 0.5) then
      obj4004C4_6_14.visibility       = #NONE
      obj4004C4_6_14_02.visibility    = #NONE
      clone4004C4_6_14.visibility     = #BOTH
      clone4004C4_6_14_02.visibility = #BOTH
      aVec2 = vector(0, 0, 0.05)
      fPos2 = vector(xDistance * 1, 0, 0)
      animateSubAssembly(clone4004C4_6_14, -aVec2, fPos2)
      animateSubAssembly(clone4004C4_6_14_02, -aVec2, -fPos2)
      sprite(spriteList.getAt(11)).loc = changedPos
      member("4004C4-6-14 Text").color = changedColor
    end if
    if clone4004C4_6_14.transform.position = vector(xDistance * 1, 0, 0) then
      cN4004C4_6_14.visibility        = #BOTH
      cN4004C4_6_14_02.visibility     = #BOTH
      clone4004C4_6_14.visibility     = #NONE
      clone4004C4_6_14_02.visibility = #NONE
      aVec = vector(0.05, 0, 0)
      animateSubAssembly(obj712442, -aVec, finalPos)
      animateSubAssembly(obj712442_02, aVec, finalPos)
      animateSubAssembly(cN4004C4_6_14, -aVec, finalPos)
      animateSubAssembly(cN4004C4_6_14_02, aVec, finalPos)
      sprite(spriteList.getAt(11)).loc = originalPos
      member("4004C4-6-14 Text").color = blackColor
      sprite(spriteList.getAt(26)).loc = changedPos
      member("712442 Text").color      = changedColor
    end if
    if obj712442.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj0646_0832_8, animatedVec, finalPos)
      animateSubAssembly(obj0646_0832_8_02, animatedVec, finalPos)
      animateSubAssembly(obj0646_0832_8_03, animatedVec, finalPos)
      animateSubAssembly(obj0646_0832_8_04, animatedVec, finalPos)
      sprite(spriteList.getAt(26)).loc = originalPos
      member("712442 Text").color      = blackColor
      sprite(spriteList.getAt(7)).loc  = changedPos
      member("0646-0832-8 Text").color = changedColor
    end if
  else if stepNo = 8 then
    -- ** OPTIONAL PARTS **
    -- ** NO ANIMATION **
  else if stepNo = 9 then
    animateSubAssembly(obj712172, animatedVec, finalPos) -- ** B-36 **
    if obj712172.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj706595, animatedVec, finalPos) -- ** B-35 **
      animateSubAssembly(obj4004_6_6_12_10, animatedVec, finalPos) -- ** B-34 **
      animateSubAssembly(obj4004_6_6_12_11, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_12, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_14, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_15, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_16, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_17, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_18, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_19, animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_20, animatedVec, finalPos)
    animateSubAssembly(obj4004_6_6_12_21, animatedVec, finalPos)
    animateSubAssembly(obj4004_6_6_12_22, animatedVec, finalPos)
    sprite(spriteList.getAt(4)).loc  = originalPos
    member("712172 Text").color      = blackColor
    sprite(spriteList.getAt(13)).loc = changedPos
    member("706595 Text").color      = changedColor
  end if
  if obj706595.transform.position = vector(0, 0, 0) then
    animateSubAssembly(obj706592, animatedVec, finalPos) -- ** B-33 **
    sprite(spriteList.getAt(13)).loc = originalPos
    member("706595 Text").color      = blackColor
    sprite(spriteList.getAt(12)).loc = changedPos
    member("706592 Text").color      = changedColor
  end if
  if obj706592.transform.position = vector(0, 0, 0) then
    animateSubAssembly(obj706611, -animatedVec, finalPos) -- ** B-32 **
    animateSubAssembly(obj4004_5_5_10_11, -animatedVec, finalPos) -- ** B-31 **
    animateSubAssembly(obj4004_5_5_10_12, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_13, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_14, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_15, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_16, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_17, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_18, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_19, -animatedVec, finalPos)
    animateSubAssembly(obj4004_5_5_10_20, -animatedVec, finalPos)
    sprite(spriteList.getAt(12)).loc = originalPos
    member("706592 Text").color      = blackColor
    sprite(spriteList.getAt(16)).loc = changedPos
    member("706611 Text").color      = changedColor
  end if
else if stepNo = 10 then
  animateSubAssembly(obj710511, animatedVec, finalPos) -- ** T-28 **
  if obj710511.transform.position = vector(0, 0, 0) then
    animatedVec = vector(0.05, 0, 0)
    animateSubAssembly(obj710646, animatedVec, finalPos) -- ** T-29 **
    sprite(spriteList.getAt(18)).loc = originalPos
    member("710511 Text").color      = blackColor
    sprite(spriteList.getAt(5)).loc  = changedPos
    member("710646 Text").color      = changedColor
  end if
else if stepNo = 11 then
  animateSubAssembly(obj762482, animatedVec, finalPos) -- ** T-27B **
  animateSubAssembly(obj737225, animatedVec, finalPos) -- ** T-20 **
  animateSubAssembly(obj737225_02, animatedVec, finalPos)
  animateSubAssembly(obj737225_03, animatedVec, finalPos)
  animateSubAssembly(obj737225_04, animatedVec, finalPos)
  animateSubAssembly(obj737225_05, animatedVec, finalPos)
  animateSubAssembly(obj737225_06, animatedVec, finalPos)
  animateSubAssembly(obj737225_07, animatedVec, finalPos)
  animateSubAssembly(obj737225_08, animatedVec, finalPos)
  animateSubAssembly(obj737225_09, animatedVec, finalPos)
  if obj762482.transform.position = vector(0, 0, 0) then
    animateSubAssembly(obj712006, animatedVec, finalPos) -- ** T-25 **
    sprite(spriteList.getAt(34)).loc = originalPos
    member("762482 Text").color      = blackColor
    sprite(spriteList.getAt(6)).loc  = changedPos
    member("712006 Text").color      = changedColor
  end if
  if obj712006.transform.position = vector(0, 0, 0) then
    animateSubAssembly(obj714051, animatedVec, finalPos) -- ** T-19 **
    animateSubAssembly(obj706606, animatedVec, finalPos) -- ** T-24 **
    animateSubAssembly(obj712113, animatedVec, finalPos) -- ** T-23 **
    animateSubAssembly(obj706610, animatedVec, finalPos) -- ** T-22 **
    animateSubAssembly(obj3415_90P, animatedVec, finalPos) -- ** T-21 **
    sprite(spriteList.getAt(6)).loc  = originalPos
    member("712006 Text").color      = blackColor
    sprite(spriteList.getAt(28)).loc = changedPos
    member("714051 Text").color      = changedColor
  end if
else if stepNo = 12 then
```

```
    animateSubAssembly(obj712159, -animatedVec, finalPos) -- ** T-18 **
    if obj712159.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj706595_02, -animatedVec, finalPos) -- ** T-17 **
      animateSubAssembly(obj4004_6_6_12, -animatedVec, finalPos) -- ** T-16 **
      animateSubAssembly(obj4004_6_6_12_02, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_03, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_04, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_05, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_06, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_07, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_08, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_09, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_13, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_23, -animatedVec, finalPos)
      animateSubAssembly(obj4004_6_6_12_24, -animatedVec, finalPos)
      sprite(spriteList.getAt(22)).loc = originalPos
      member("712159 Text").color      = blackColor
      sprite(spriteList.getAt(13)).loc = changedPos
      member("706595 Text").color      = changedColor
    end if
    if obj706595_02.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj712160, -animatedVec, finalPos) -- ** T-14 **
      sprite(spriteList.getAt(13)).loc = originalPos
      member("706595 Text").color      = blackColor
      sprite(spriteList.getAt(23)).loc = changedPos
      member("712160 Text").color      = changedColor
    end if
    if obj712160.transform.position = vector(0, 0, 0) then
      animateSubAssembly(objNAS1351_3H8, animatedVec, finalPos) -- ** T-15 **
      animateSubAssembly(objNAS1351_3H8_02, animatedVec, finalPos)
      animateSubAssembly(objNAS1351_3H8_03, animatedVec, finalPos)
      animateSubAssembly(objNAS1351_3H8_04, animatedVec, finalPos)
      sprite(spriteList.getAt(23)).loc = originalPos
      member("712160 Text").color      = blackColor
      sprite(spriteList.getAt(3)).loc  = changedPos
      member("NAS1351-3H8 Text").color = changedColor
    end if
    if objNAS1351_3H8.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj706611_02, animatedVec, finalPos) -- ** T-13 **
      animateSubAssembly(obj4004_5_5_10, animatedVec, finalPos) -- ** T-12 **
      animateSubAssembly(obj4004_5_5_10_02, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_03, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_04, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_05, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_06, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_07, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_08, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_09, animatedVec, finalPos)
      animateSubAssembly(obj4004_5_5_10_10, animatedVec, finalPos)
      sprite(spriteList.getAt(3)).loc  = originalPos
      member("NAS1351-3H8 Text").color = blackColor
      sprite(spriteList.getAt(16)).loc = changedPos
      member("706611 Text").color      = changedColor
    end if
  else if stepNo = 13 then
    animateSubAssembly(objNAS1523AA6E, -animatedVec, finalPos) -- ** M-6 **
    animateSubAssembly(objNAS1523AA6E_02, animatedVec, finalPos)
    animateSubAssembly(objNAS1523AA6E_03, -animatedVec, finalPos)
    animateSubAssembly(objNAS1523AA6E_04, animatedVec, finalPos)
    if objNAS1523AA6E.transform.position = vector(0, 0, 0) then
      animateSubAssembly(objMS15795_810, animatedVec, finalPos) -- ** M-5 **
      animateSubAssembly(objMS15795_810_02, animatedVec, finalPos)
      sprite(spriteList.getAt(39)).loc = originalPos
      member("NAS1523AA6E Text").color = blackColor
      sprite(spriteList.getAt(35)).loc = changedPos
      member("MS15795-810 Text").color = changedColor
    end if
    if objMS15795_810.transform.position = vector(0, 0, 0) then
      animateSubAssembly(objMS21043_4, animatedVec, finalPos) -- ** M-4 **
      animateSubAssembly(objMS21043_4_02, animatedVec, finalPos)
      sprite(spriteList.getAt(35)).loc = originalPos
      member("MS15795-810 Text").color = blackColor
      sprite(spriteList.getAt(37)).loc = changedPos
      member("MS21043-4 Text").color   = changedColor
    end if
    if objMS21043_4.transform.position = vector(0, 0, 0) then
      animateSubAssembly(obj710626, -animatedVec, finalPos) -- ** M-3 **
      animateSubAssembly(obj710626_02, -animatedVec, finalPos)
      sprite(spriteList.getAt(37)).loc = originalPos
      member("MS21043-4 Text").color   = blackColor
      sprite(spriteList.getAt(1)).loc  = changedPos
      member("710626 Text").color      = changedColor
    end if
  else if stepNo = 14 then
    animateSubAssembly(cL739500, -animatedVec, finalPos) -- ** TOP ASSEMBLY **
    if cL739500.transform.position = vector(0, 0, 0) then
      animateSubAssembly(cL712169, animatedVec, finalPos) -- ** BOTTOM ASSEMBLY **
      sprite(spriteList.getAt(32)).loc = originalPos
      member("739500 Text").color      = blackColor
      sprite(spriteList.getAt(25)).loc = changedPos
      member("712169 Text").color      = changedColor
    end if
    if cL712169.transform.position = vector(0, 0, 0) then
      animateSubAssembly(cENAS1351_3_16, -animatedVec, finalPos) -- ** SCREW **
      sprite(spriteList.getAt(25)).loc  = originalPos
      member("712169 Text").color       = blackColor
      sprite(spriteList.getAt(2)).loc   = changedPos
      member("NAS1351-3-16 Text").color = changedColor
    end if
    if cENAS1351_3_16.transform.position = vector(0, 0, 0) then
      animateSubAssembly(cEMS21043_3, animatedVec, finalPos) -- ** SELF-LOCKING NUT **
      sprite(spriteList.getAt(2)).loc   = originalPos
      member("NAS1351-3-16 Text").color = blackColor
      sprite(spriteList.getAt(36)).loc  = changedPos
      member("MS21043-3 Text").color    = changedColor
    end if
  end if

end
-- ************************************************
```

APPENDIX B: LINGO SCRIPT FOR INITIALIZATION

```
-- The "initializePosition" function initializes the position
-- for each individual part inside the assembly 3D cast member
-- and also each individual part inside 3D cast member associated
-- with the secondary window. This function sets the default position
-- of the main window's camera and boundary box as well.
on initializePosition

  -- ** CAMERA SCENE **
  sceneCamera = obj.camera("DefaultView")
  --  sceneCamera.transform.position = vector(0.81, -3.80, 2.60)
  --  sceneCamera.transform.rotation = vector(56, 20.50, 0)

  -- ** MAIN OBJECTS **
  --  boundaryBox.transform.rotation = basePos
  -- ** FULL ASSEMBLY (120 OBJECTS) **
  repeat with i = 1 to 120
    objList.getAt(i).transform.position = basePos
  end repeat
  -- ** FULL ASSEMBLY **

  -- ** CLONE OBJECTS OUT OF SCREEN **
  repeat with i = 1 to 120
    cloneList.getAt(i).transform.position = offPos
  end repeat

  -- ** EXTRA CLONE FOR SIMULATION **
  cN4004C4_6_14.transform.position    = offPos
  cN4004C4_6_14_02.transform.position = offPos
  cN712442.transform.position         = offPos
  cN712442_02.transform.position      = offPos
  cL739497.transform.position         = offPos
  cX739497.transform.position         = offPos
  cL714051.transform.position         = offPos
  cX714051.transform.position         = offPos
  cLNAS1351_3_16.transform.position   = offPos
  cLMS21043_3.transform.position      = offPos
  cL712169.transform.position         = offPos
  cL739500.transform.position         = offPos
  cE712169.transform.position         = offPos
  cE739500.transform.position         = offPos
  cE712165_1.transform.position       = offPos
  cX712169.transform.position         = offPos
  cX739500.transform.position         = offPos
  cENAS1351_3_16.transform.position   = offPos
  cEMS21043_3.transform.position      = offPos
  cL706595.transform.position         = offPos
  cX706595.transform.position         = offPos
  cL706611.transform.position         = offPos
  cX706611.transform.position         = offPos
  cL706595_02.transform.position      = offPos
  cX706595_02.transform.position      = offPos
  cL706611_02.transform.position      = offPos
  cX706611_02.transform.position      = offPos
  cL762482.transform.position         = offPos
  cX762482.transform.position         = offPos

  -- ** SUB OBJECTS **
  sub710626.transform.position       = vector(1.03, -1.91, 2.95)
  sub710626.transform.rotation       = vector(40, 27.50, 0)
  subNAS1351_3_16.transform.position = vector(0.55, -1.28, 2.27)
  subNAS1351_3_16.transform.rotation = vector(36.50, 18.00, 0)
  subNAS1351_3H8.transform.position  = vector(0.55, -1.28, 2.27)
  subNAS1351_3H8.transform.rotation  = vector(36.50, 18.00, 0)
  sub712172.transform.position       = vector(1.04, -3.81, 5.24)
  sub712172.transform.rotation       = vector(46, 16, 0)
  sub710646.transform.position       = vector(0.42, -0.97, 0.78)
  sub710646.transform.rotation       = vector(49.5, 31.5, 0)
```

```
    sub712006.transform.position       = vector(0.29, -2.55, 1.69)
    sub712006.transform.rotation       = vector(58, 11.5, 0)
    sub0646_0832_8.transform.position  = vector(0.59, -1.14, 0.63)
    sub0646_0832_8.transform.rotation  = vector(47.5, 51, 0)
    sub3415_90P.transform.position     = vector(-0.76, -2.11, -1.95)
    sub3415_90P.transform.rotation     = vector(-136, -23.5, -177.5)
    sub4004_5_5_10.transform.position  = vector(0.28, -1.01, 0.82)
    sub4004_5_5_10.transform.rotation  = vector(52.5, 21, 0)
    sub4004_6_6_12.transform.position  = vector(0.4, -0.88, 1.03)
    sub4004_6_6_12.transform.rotation  = vector(41.5, 23.5, 0)
    sub4004C4_6_14.transform.position  = vector(0.19, -0.89, 0.76)
    sub4004C4_6_14.transform.rotation  = vector(50.5, 15, 0)
    sub706592.transform.position       = vector(0.55, -2.01, 1.88)
    sub706592.transform.rotation       = vector(48.5, 18, 0)
    sub706595.transform.position       = vector(0.44, -1.84, 1.63)
    sub706595.transform.rotation       = vector(51, 17, 0)
    sub706606.transform.position       = vector(0.33, -1.53, 1.15)
    sub706606.transform.rotation       = vector(52.5, 16.5, 0)
    sub706610.transform.position       = vector(0.43, -1.55, 1.1)
    sub706610.transform.rotation       = vector(53, 22, 0)
    sub706611.transform.position       = vector(0.37, -1.19, 1.48)
    sub706611.transform.rotation       = vector(40.5, 15.5, 0)
    sub706620.transform.position       = vector(0.71, -2.40, 1.89)
    sub706620.transform.rotation       = vector(50, 20.50, 0)
    sub710511.transform.position       = vector(3.3, 0.43, -1.85)
    sub710511.transform.rotation       = vector(-134.5, -38, -51)
    sub710641.transform.position       = vector(0.21, -1.14, 0.82)
    sub710641.transform.rotation       = vector(56, 15.5, 0)
    sub711430.transform.position       = vector(0.83, -1.3, 0.72)
    sub711430.transform.rotation       = vector(55.5, 36.5, 0)
    sub712113.transform.position       = vector(-0.64, -1.26, 0.71)
    sub712113.transform.rotation       = vector(52, -56.5, 0)
    sub712159.transform.position       = vector(0.7, -3.27, 3.81)
    sub712159.transform.rotation       = vector(54, 16.5, 0)
    sub712160.transform.position       = vector(-0.72, -3.56, -1.65)
    sub712160.transform.rotation       = vector(-118.5, -19.5, 180)
    sub712165_1.transform.position     = vector(0.54, -3.30, 3.53)
    sub712165_1.transform.rotation     = vector(40.5, 10.5, 0)
    sub712169.transform.position       = vector(-2.83, -4.46, -2)
    sub712169.transform.rotation       = vector(135, 53.5, 0)
    sub712442.transform.position       = vector(0.63, -1.2, 1.17)
    sub712442.transform.rotation       = vector(55, 32.5, 0)
    sub713737.transform.position       = vector(0.46, -1.37, 0.64)
    sub713737.transform.rotation       = vector(53, 50, 0)
    sub714051.transform.position       = vector(0.56, -2.28, 2.43)
    sub714051.transform.rotation       = vector(47.5, 17, 0)
    sub726735.transform.position       = vector(0.31, -1.7, 1.09)
    sub726735.transform.rotation       = vector(57, 16.5, 0)
    sub737225.transform.position       = vector(-0.51, 2.15, 0.96)
    sub737225.transform.rotation       = vector(-43, -46.5, -20)
    sub739497.transform.position       = vector(1.44, -3, 3.3)
    sub739497.transform.rotation       = vector(44, 23.5, 0)
    sub739500.transform.position       = vector(-0.8, -1.73, -2.69)
    sub739500.transform.rotation       = vector(136, 16.5, 0)
    sub740981.transform.position       = vector(1.59, -1.17, 0.11)
    sub740981.transform.rotation       = vector(-40.5, -86, -180)
    sub762482.transform.position       = vector(0.79, -4.25, 2.56)
    sub762482.transform.rotation       = vector(58, 17.5, 0)
    subMS15795_810.transform.position  = vector(0.28, -1.60, 1.07)
    subMS15795_810.transform.rotation  = vector(56, 15, 0)
    subMS21043_4.transform.position    = vector(0.40, -6.09, 2.52)
    subMS21043_4.transform.rotation    = vector(96, 58, 0)
    subMS21043_06.transform.position   = vector(0.27, -0.96, 0.85)
    subMS21043_06.transform.rotation   = vector(50.5, 19.5, 0)
    subNAS1523AA6E.transform.position  = vector(0.30, -1.61, 1.05)
    subNAS1523AA6E.transform.rotation  = vector(56.5, 16.5, 0)

end
-- **************************************************
```

APPENDIX C: LINGO SCRIPT FOR COMPUTATION

```
global totalTime, isFinished
global positionA, positionB, rBVector
-- ************************************************
on calculateRBVector

  rBVector = positionB.worldPosition

end
-- ************************************************
on computeVelocity theT

  -- ** USER INPUT **
  -- value(member("horizontalAxisOmega").text) = Rotating about horizontal axis (Arm)
  -- value(member("verticalAxisAcc").text) = Angular Acc. about z axis
  zOmega = 0.25 -- fix value
  zAlpha = value(member("verticalAxisAcc").text)
  zOmega = zOmega + (zAlpha * theT)

  xOmega = 0.4  -- fix value
  xAlpha = value(member("horizontalAxisAcc").text)
  xOmega = xOmega + (xAlpha * theT)

  omega  = vector(- xOmega, 0, zOmega)

  -- ** EQUATION NO. 6 **
  -- Based on the velocity principle
  vB          = omega.cross(rBVector)
  vBMangitude = vB.magnitude

  the floatPrecision = 2
  -- ** SHOW OUTPUT **
  -- vB(i)     = member("vBx").text
  -- vB(j)     = member("vBy").text
  -- vB(k)     = member("vBz").text
  -- Vel.Mag. = member("vMag").text
  member("vBx").text  = string(vB.x)
  member("vBy").text  = string(vB.y)
  member("vBz").text  = string(vB.z)
  member("vMag").text = string(vBMangitude)

end
-- ************************************************
on computeAcc theT
  -- ** USER INPUT **
  -- value(member("horizontalAxisOmega").text) = Rotating about horizontal axis (Arm)
  -- value(member("verticalAxisAcc").text) = Angular Acc. about z axis
  zOmega = 0.25 -- fix value
  zAlpha = value(member("verticalAxisAcc").text)
  zOmega = zOmega + (zAlpha * theT)

  xOmega = 0.4  -- fix value
  xAlpha = value(member("horizontalAxisAcc").text)
  xOmega = xOmega + (xAlpha * theT)

  alphaAcc = vector(-xAlpha, 0, zAlpha)
  omega    = vector(- xOmega, 0, zOmega)
  omegaRB  = omega.cross(rBVector)

  -- ** EQUATION NO. 7 **
  -- Based on the acceleration principle
  aB          = alphaAcc.cross(rBVector) + omega.cross(omegaRB)
  aBMangitude = aB.magnitude

  the floatPrecision = 2
  -- ** SHOW OUTPUT **
  -- aB(i)     = member("aBx").text
  -- aB(j)     = member("aBy").text
  -- aB(k)     = member("aBz").text
  -- Acc.Mag. = member("aMag").text
  member("aBx").text  = string(aB.x)
  member("aBy").text  = string(aB.y)
  member("aBz").text  = string(aB.z)
  member("aMag").text = string(aBMangitude)

end
-- ***************************************************
```

APPENDIX D: LINGO SCRIPT FOR SIMULATION

```
global isFinished
global totalTime
global positionA, positionB, connectionAB, rodLength, sceneCamera, boundaryBox
global CurTime
global zOmega, xOmega
global initialAngle
-- ************************************************
on initialStep

  totalTime  = 1.95 -- to make total 2.0
  zOmega     = 0.25 -- fix value
  xOmega     = 0.4  -- fix value
  isFinished = false
  CurTime    = 0.00

  member("time").text = "0.0"
  member("vBx").text  = "0.00"
  member("vBy").text  = "0.00"
  member("vBz").text  = "0.00"
  member("vMag").text = "0.00"

  member("aBx").text  = "0.00"
  member("aBy").text  = "0.00"
  member("aBz").text  = "0.00"
  member("aMag").text = "0.00"

  create3DWorld()
  initialBar()

end
-- ************************************************
on startBar

  sprite(34).locH = 405 + 5
  sprite(35).locH = 367 + 5
  sprite(36).locH = 374 + 5 -- from 374.6

  sprite(34).moveableSprite = TRUE
  sprite(35).moveableSprite = TRUE
  sprite(36).moveableSprite = TRUE

end
-- ************************************************
on stopBar

  sprite(34).moveableSprite = FALSE
  sprite(35).moveableSprite = FALSE
  sprite(36).moveableSprite = FALSE

end
-- ************************************************
on initialBar

  minSlider = 365 + 5
  maxSlider = 425 + 5

  sprite(34).locV = 270 - 10
  sprite(35).locV = 310 - 10
  sprite(36).locV = 350 - 10

  repeat with spriteNo = 34 to  36

    if the locH of sprite spriteNo < minSlider then
      set the locH of sprite spriteNo to minSlider
    end if
    if the locH of sprite spriteNo > maxSlider then
      set the locH of sprite spriteNo to maxSlider
    end if

  end repeat

  the floatPrecision = 1

  bar1Location = abs(sprite(34).locH - minSlider)
  member("initialAngle").text = string(20 + bar1Location)
  bar2Location = float(abs(sprite(35).locH - minSlider))
  member("horizontalAxisAcc").text = string(bar2Location / 20)
  bar3Location = float(abs(sprite(36).locH - minSlider))
  member("verticalAxisAcc").text = string(bar3Location / 12)

end
-- *************************************************
on animate

  if (isFinished = true) then
    return
  end if

  animate3DWorld()
  updateStage

end
-- *************************************************
on animate3DWorld

  if CurTime >= totalTime then
    isFinished = true
  end if

  CurTime = CurTime + 0.05
  member("time").text = string(CurTime)

  zAlpha      = value(member("verticalAxisAcc").text)
  theZDegree = (zOmega * (180/PI) * .05) + (.5 * zAlpha * (180/PI) * .05 * .05)
  zOmega      = zOmega + (zAlpha * 0.05)
  connectionAB.rotate(0, 0, theZDegree, positionA)

  xAlpha      = value(member("horizontalAxisAcc").text)
  theXDegree = (xOmega * (180/PI) * .05) + (.5 * xAlpha * (180/PI) * .05 * .05)
  xOmega      = xOmega + (xAlpha * 0.05)
  connectionAB.rotate(-theXDegree, 0, 0, positionA)

  calculateRBVector()
  computeVelocity(CurTime)
  computeAcc(CurTime)

end
-- *************************************************
```

APPENDIX E: LINGO SCRIPT FOR 3D WORLD CREATION

```
global positionA, positionB, connectionAB, sceneCamera, boundaryBox, cloneA
global rodLength
global initialAngle
-- *************************************
on create3DWorld

  -- ***********************************************
  -- initialize 3D world
  -- ***********************************************
  s = member("Blank Scene")
  s.resetWorld()

  sceneCamera = s.camera[1]

  -- ***********************************************
  -- create shaders
  -- ***********************************************
  -- ** BLACK COLOR **
  textureBlack = s.newTexture("textureBlack", #fromCastmember, member("black"))
  shaderBlack  = s.newShader("shaderBlack",#standard)
  shaderBlack.texture = textureBlack

  -- ** BLUE COLOR **
  textureBlue = s.newTexture("textureBlue", #fromCastmember, member("blue"))
  shaderBlue  = s.newShader("shaderBlue",#standard)
  shaderBlue.texture = textureBlue

  -- ** RED COLOR **
  textureRed = s.newTexture("textureRed", #fromCastmember, member("red"))
  shaderRed  = s.newShader("shaderRed",#standard)
  shaderRed.texture = textureRed

  -- ***********************************************
  -- create models
  -- ***********************************************
  -- ** AXIS AND CONNECTION MODEL RESOURCE **
  rod = s.newModelResource("rod", #cylinder)
  s.modelResource("rod").topRadius    = 1.0
  s.modelResource("rod").bottomRadius = 1.0
  s.modelResource("rod").height       = 40 -- from 35
  rodLength = s.modelResource("rod").height

  -- ** AEROW HEAD MODEL RESOURCE **
  cone = s.newModelResource("cone", #cylinder)
  s.modelResource("cone").topRadius    = 0
  s.modelResource("cone").bottomRadius = 3
  s.modelResource("cone").height       = 3.5
  coneLength = s.modelResource("cone").height

  -- ** Y AXIS **
  m1 = s.newModel("rod1", rod)
  m1.shaderlist = shaderBlack
  c1 = s.newModel("cone1", cone)
  c1.transform.position = vector(0, (rodLength / 2) + (coneLength / 2), 0)
  c1.shaderlist = shaderBlack
  m1.addchild(c1)

  m1.transform.scale    = vector(0.75, 1.25, 0.75)
  m1.transform.position = vector(0, 0, 0)

  -- ** X AXIS **
  m2 = s.newModel("rod2", rod)
  m2.shaderlist = shaderBlue
  c2 = s.newModel("cone2", cone)
  c2.transform.position = vector(0, (rodLength / 2) + (coneLength / 2), 0)
  c2.shaderlist = shaderBlue
  m2.addchild(c2)
```

```
m2.transform.scale    = vector(0.75, 1.75, 0.75) -- from 1.25
m2.transform.position = vector(0, 0, 0)
m2.rotate(0, 0, -90, #world)

-- ** Z AXIS **
m3 = s.newModel("rod3", rod)
m3.shaderlist = shaderRed
c3 = s.newModel("cone3", cone)
c3.transform.position = vector(0, (rodLength / 2) + (coneLength / 2), 0)
c3.shaderlist = shaderRed
m3.addchild(c3)

m3.transform.scale    = vector(0.75, 1.25, 0.75)
m3.transform.position = vector(0, 0, 0)
m3.rotate(90, 0, 0, #world)

-- ** AXIS GROUP **
m1.addChild(m2)
m1.addChild(m3)

-- ** OBJECT MODEL RESOURCE **
ball = s.newModelResource("ball", #sphere)
s.modelResource("ball").radius = 3

-- ** OBJECT A **
positionA = s.newModel("positionA", ball)
positionA.shaderlist = shaderBlack
positionA.transform.position = vector(0, 0, 0)
cloneA = positionA.clone("cloneA")

-- ** CONNECTION BETWEEN A AND B **
connectionAB = s.newModel("connectionAB", rod)
connectionAB.shaderlist = shaderRed
connectionAB.transform.position = vector(0, rodLength / 2, 0)
connectionAB.rotate(90, 0, 0, #world)

-- ** OBJECT B **
positionB = s.newModel("positionB", ball)
positionB.shaderlist = shaderBlack
positionB.transform.position = vector(0, 0, rodLength) -- from rodLength * 75

connectionAB.addChild(positionB)
cloneA.addChild(connectionAB)
positionA.addChild(cloneA)
initialAngle = value(member("initialAngle").text)
cloneA.rotate(0, -initialAngle, 0, #world) -- initial angle is 30 degree

-- ************************************************
-- create texts
-- ************************************************
-- ** Z **
textZ = member("z")
mr = textZ.extrude3D(s)
mr.tunnelDepth = 2.5
mZ = s.newModel("z", mr)
mZ.shaderList = shaderRed
mZ.transform.scale    = vector(0.5, 0.5, 0.5)
mZ.transform.position = vector(0, -10, (rodLength / 2) + (coneLength / 2) + 5)
mZ.rotate(90, 0, -180)

-- ** X **
textX = member("x")
mr = textX.extrude3D(s)
mr.tunnelDepth = 2.5
mX = s.newModel("x", mr)
mX.shaderList = shaderBlue
mX.transform.scale    = vector(0.5, 0.5, 0.5)
mX.transform.position = vector(rodLength + coneLength, 0, -5)
mX.rotate(-90, 0, -180)
```

```
  -- ** Y **
  textY = member("y")
  mr = textY.extrude3D(s)
  mr.tunnelDepth = 2.5
  mY = s.newModel("y", mr)
  mY.shaderList = shaderBlack
  mY.transform.scale    = vector(0.5, 0.5, 0.5)
  mY.transform.position = vector(0, (rodLength / 2) + (coneLength / 2) + 5, -10)
  mY.rotate(90, 0, 0)

  -- ** AXIS GROUP **
  m1.addChild(mZ)
  m1.addChild(mX)
  m1.addChild(mY)

  -- ** BOUNDARY BOX **
  dimension = 45
  cube = s.newModelResource("box", #box)
  s.modelResource("box").height = dimension
  s.modelResource("box").width  = dimension
  s.modelResource("box").length = dimension
  boundaryBox = s.newModel("boundaryBox", cube)
  boundaryBox.visibility = #NONE

  -- ** SCENE GROUP **
  m1.addChild(positionA)
  boundaryBox.addChild(m1)

end
-- ****************************************
```

APPENDIX F: ASP SCRIPT FOR USER VERIFICATION

```
<!-- #include file="../includes/header.asp"-->

<%
'***************************************************************************
'File Name: login.asp
'Description:
'This file is to check whether a user is allowed to enter the system as well
'as to clear any particular information about the user when he is leaving.
'The following information are stored in cookies:
'      "userName", "userType", "firstName", "lastName"
'***************************************************************************

Dim str_currentUser, str_currentPass, str_currentType
str_currentUser = Request.Form("userName")
str_currentPass = Request.Form("password")
str_currentType = "0"        'default guest type

If str_currentUser = "" OR str_currentPass = "" Then
      Response.Redirect "../error/error_id.htm"
End If

'Variables for the database
'--------------------------
Dim obj_connect
Set obj_connect = Server.CreateObject("ADODB.Connection")
obj_connect.Open "DSN=" & Application("tinkerDB") & ";UID=" & Application("userDB") &
";password=" & Application("passDB")

Dim str_first, str_last, str_password

If Not isGuest() Then
      getTinkerInfo()
End If

'Verifying the password
'----------------------
If str_password <> str_currentPass Then
      Response.Redirect "../error/error_id.htm"
End If

'Store info into cookies
'-----------------------
Response.Cookies("userName") = str_currentUser
Response.Cookies("userType") = str_currentType
Response.Cookies("firstName") = str_first
Response.Cookies("lastName") = str_last
Response.Cookies("currentModule").Expires = #8/8/1998#
Response.Cookies("currentLessons").Expires = #8/8/1998#

If str_currentType = "3" OR str_currentType = "2" Then
      Response.Redirect "../application/home_admin.asp"
Else
      Response.Redirect "../application/home.asp"
End If

obj_connect.Close
Set obj_connect = Nothing

'***************************************************************************
Function isGuest()

      isGuest = False

      Dim obj_retrieve, str_SQL
      str_SQL = "SELECT password, fname, lname FROM Guest WHERE username = '" &
str_currentUser & "';"
      Set obj_retrieve = obj_connect.Execute(str_SQL)
```

```
	If obj_retrieve.EOF = False Then
		isGuest = True
		str_first = obj_retrieve("fname")
		str_last = obj_retrieve("lname")
		str_password = obj_retrieve("password")
	End If

End Function
'***************************************************************************

'***************************************************************************
Function getTinkerInfo()

	Dim obj_retrieve, str_SQL
	str_SQL = "SELECT password, type, fname, lname FROM TinkerUser WHERE username = '"
& str_currentUser & "';"
	Set obj_retrieve = obj_connect.Execute(str_SQL)

	If obj_retrieve.EOF = True Then
		Response.Redirect "../error/error_id.htm"
	End If

	str_first = obj_retrieve("fname")
	str_last = obj_retrieve("lname")
	str_password = obj_retrieve("password")
	str_currentType = CStr(obj_retrieve("type"))

End Function
'***************************************************************************
%>
```

APPENDIX G: SUMMARY OF IMPLEMENTED TECHNOLOGIES

Technologies	Capabilities
Macromedia Dreamweaver	▪ Visual HTML editor ▪ Generate HTML files
Macromedia Director	▪ Develop 2D and 3D graphics-oriented applications ▪ Compress complex 2D and 3D applications ▪ Support a built-in programming language (Lingo) ▪ Lingo is used to design the navigation system, presentations, and simulations ▪ Communication with the web-based database ▪ Support interactive 3D graphics ▪ Allows the delivery of high-performance 3D visualization to the Internet world ▪ 3D model can be created and controlled using only Lingo ▪ Allow 3D object to be imported from other 3D applications such as 3ds Max, Swift 3D, etc.
Macromedia Flash	▪ Vector-based animation tool ▪ Develop high-quality animation at low file size ▪ Compress high-resolution photograph and image
Adobe Photoshop	▪ Develop graphics ▪ Optimize high-resolution photograph
Sound Forge	▪ Edit audio instruction and sound
Microsoft SQL Server	▪ Manage web-based database system ▪ Provide storage for a large number of users ▪ Support large number of users concurrently
Structure Query Language (SQL)	▪ Communicate with a database ▪ Language for querying a database ▪ Retrieve and remove data from a database ▪ Add data to a database ▪ Modify data in a database
ASP Scripts	▪ Provide server-based interactivity ▪ Communicate with the database ▪ Create dynamic page
VBScript	▪ Develop ASP
JavaScript	▪ Add functionality and interactivity to HTML
Swift3D	▪ Develop intermediate-level 3D applications ▪ 3D application is a small file size (Flash movies) ▪ 3D application can be presented on the web ▪ Easy to use
3ds Max	▪ Develop high-level 3D applications ▪ 3D application is a small file size (Flash movies) ▪ 3D application can be presented on the web ▪ Support w3d format for Director