A SCORM COMPLIANT COURSE FRAMEWORK FOR DISTANCE LEARNING

by

Barış İbrahim SÖNMEZER

B.S. in C.E., Istanbul Technical University, 2000

152608

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

152408

A SCORM COMPLIANT COURSE FRAMEWORK
FOR DISTANCE LEARNING

APPROVED BY:

Dr. Haluk Bingöl ....
(Thesis Supervisor)

Prof. Emin Anarım ....

Dr. Bülent Yağcı ....

DATE OF APPROVAL: 17.06.2004

ACKNOWLEDGEMENT

First, I'd like to thank my family for putting up with me during the time this thesis was being produced. It has taken me away from them more than I'd like to admit.

A big "thank you" goes out to my colleagues at İTÜ Computer Center for their patience during the writing of the thesis. I couldn't have asked for a finer team.

I'd like to give special recognition to my advisor Haluk Bingöl for providing his expert knowledge during the thesis. He's not afraid to tell me when I'm wrong.

I would also like to thank Ozan Önder Özener for his help in web interface design.

And last a thank you to my best friend, Selin, who always encourages me to complete the thesis and spent hours at the computer with me during the writing part of this book.

ABSTRACT

A SCORM COMPLIANT COURSE FRAMEWORK FOR DISTANCE LEARNING

The purpose of the thesis is to construct a distance learning system for the educational content of the CMPE 150 course named "Introduction to Computing". Actually, a framework has been developed to transport other departmental courses to the electronic media.

The whole content is constructed from the beginning. Seven subjects are chosen for delivery, which are about the basic concepts in computer engineering. The aim is to give these basics to the new computer engineer candidates.

The electronic content is prepared using advanced web technologies. As the Internet is widespread used in the World, WWW is the best method for the delivery of the electronic content.

The content should be supported with visual material to attract the attention of the audience. The content, supported with interactive animations, will be more understandable and more striking. Macromedia Flash is used to construct the interactive animations for the course content.

The content should also be compatible with Learning Management Systems developed by different vendors to reach more people and implement reusability. To make the content compatible with LMS software developed by different vendors, the electronic content is organized as Sharable Content Object Reference Model (SCORM) compliant.

ÖZET

UZAKTAN EĞİTİM İÇİN SCORM UYUMLU İÇERİK SİSTEMİ

Hazırlanan projenin amacı, CMPE 150 kodlu "Bilgiişleme Giriş" isimli ders içeriği için, bir uzaktan eğitim sistemi kurulmasıdır. Aslında, diğer bölüm derslerini de elektronik ortama aktarmak amacıyla bir model yaratılmıştır.

Dersin tüm içeriği baştan oluşturulmuştur. Yedi tane konu belirlenmiş ve ders bu konular üzerine hazırlanmıştır. Konuların seçiminde amaç, yeni bilgisayar mühendisi adaylarına, bilgisayar mühendisliği ile ilgili temel kavramların anlatılmasıdır.

Ders içeriğinin elektronik ortama aktarılması, gelişmiş web teknolojileri kullanılarak gerçekleştirilmiştir. Dünya üzerinde Internet'in yaygınlığı düşünüldüğünde, hazırlanan ders içeriğinin sunumunda, World Wide Web'in araç olarak kullanılması uygundur.

Eğitim için hazırlanan içeriği, etkileşimli görüntüler katarak zenginleştirmek, içeriği takip eden kişinin dikkatini arttırmaktadır. Etkileşimli görüntüler ile desteklenmiş elektronik içerik, daha anlaşılır ve dikkat çekici bir hal almaktadır. Etkileşimli görüntülerin hazırlanmasında Macromedia Flash kullanılmıştır.

İçeriğin, farklı üreticiler tarafından geliştirilmiş olan Öğrenim Yönetim Sistemleri ile uyumlu çalışabilmesi, daha fazla kişiye ulaşması ve tekrar kullanılabilir olması açısından önemlidir. Farklı üreticiler tarafından hazırlanan Öğrenim Yönetim Sistemleri ile çalışabilmesini sağlamak amacıyla, hazırlanan içerik Sharable Content Object Reference Model (SCORM) ile uyumlu hale getirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADL	Advanced Distributed Learning
AICC	Aviation Industry Computer-Based Training Committee
API	Application Programming Interface
ARIADNE	Alliance for Remote Instructional Authoring and Distribution Networks for Europe
ASP	Active Server Pages
CAM	Content Aggregation Model
CBT	Computer Based Training
CGI	Common Gateway Interface
DHTML	Dynamic Hypertext Markup Language
DL	Distance Learning
DMA	Direct Memory Access
DNS	Domain Name System
DoD	Department of Defense
DOS	Disk Operating System
ECMA	European Computer Manufacturers Association
FDD	Floppy disk drive
FPS	Frame-per-Second
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GUI	Graphical User Interface
HR	Human Resource
HTML	Hypertext Markup Language
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
IMS	Instructional Management Systems
IPSec	IP Security
IRQ	Interrupt Request
ISO	International Organization for Standardization
IT	Information Technology

JPG	Joint Photographic Experts Group
LAN	Local Area Network
LMS	Learning Management System
LOM	Learning Object Metadata
Mbps	Mega-bit per second
NIC	Network Interface Card
ODBC	Open Database Connectivity
OOP	Object-oriented Programming
OS	Operating System
OSI	Open System Interconnection
PIF	Package Interchange File
PHP	PHP: Hypertext Preprocessor
RAM	Random Access Memory
RAS	Reliability, availability, and service ability
RJ45	Registered Jack 45
RTE	Run-Time Environment
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SSH	Secure SHell
SVGA	Super Video Graphics Array
SWF	ShockWave Flash Object
TCP/IP	Transmission Control Protocol/Internet Protocol
TOC	Table of Contents
UI	User Interface
USB	Universal Serial Bus
UTP	Unshielded Twisted Pair
VBScript	Visual Basic Script
VGA	Video Graphics Array
WAN	Wide Area Network
WWW	World Wide Web
XGA	Extended Graphics Array
XML	Extensible Markup Language

1. THESIS INTRODUCTION

1.1. Statement of the Problem

With the rapid growth of the World Wide Web (WWW) in recent years, many educators are now trying to improve the quality of their instruction by providing interactive, web-based course material to their students. In this thesis, a web-based distance learning system is implemented for the educational content of the CMPE 150 course named "Introduction to Computing".

In today's Information Age, learning is no longer restricted within the four walls of a classroom. The instructor, equipped with a textbook, is no longer the only source of educational knowledge. Information resources are everywhere, often separated from the learner by time and space. Distance learning (DL) defines the process of connecting learners with these remote resources (Yoakam, 1996). While correspondence courses have existed since the turn of the century, DL has evolved from a static, print-based delivery medium to today's interactive multimedia and online learning environments (Homan, 1997). Recently, distance learning on a Web-based environment has been receiving more and more attention (Sun *et al.*, 2001).

Many educators ask if distant students learn as much as students receiving traditional face-to-face instruction. Research comparing distance education to traditional face-to-face instruction indicates that teaching and studying at a distance can be as effective as traditional instruction, when the method and technologies used are appropriate to the instructional tasks, there is student-to-student interaction, and when there is timely teacher-to- student feedback (Verduin *et al*, 1991).

There are many advantages to deliver the interactive course content through the web. Some of them include remote accessibility to materials, support for distance learning, the development of system independent applications, and the use of hypermedia (Naps *et al.*, 1997). Web-based education is fast becoming a popular alternative to traditional face-to-face classroom based education due to limitations on student's time and ability to travel to and from class (Rahm *et al*, 1997).

The electronic content prepared for the thesis is supported with interactive animations. The WWW supports the development of interactive teaching material in which students "learn-by-doing" rather than just reading and memorizing (Schank, 1994). Through the use of multimedia development tools such as Flash and Java, educators can now create material that engages students to learn through active participation.

For reusability, the electronic content is made Sharable Content Object Reference Model (SCORM) compliant. The WWW provides a suitable means for reuse and sharing of educational materials. This change opens the possibility for educators to reuse existing materials rather than redeveloping something that already exists. It also allows educators to develop new materials and make these immediately available to others (Tsichritzis, 1999).

The preparation of the thesis consists of five main steps:

First step is to decide which subjects will be included in the course content. The electronic content is prepared for computer engineering freshmen, so the subjects should teach them the basic concepts of computer engineering. Seven subjects are chosen for this purpose. These are:

i. Information Technology Basics
ii. Computer Basics
iii. Operating Systems Basics
iv. Networking Basics
v. Internet Basics
vi. WWW Basics
vii. Network Security

Second step is about the web interface. Web pages are designed to submit the electronic content. The purpose of the web page design is a user-friendly interface which allows the users to follow course content and animations at the same screen and to navigate easily through the course content.

Third step is making research about the selected subjects. Many resources are investigated, including books, magazines, papers and the Internet. During this research, subtopics of the selected seven chapters are also decided.

Fourth step is about the interactive animations. In order to capture the attention of the audience, visual material should be included to the electronic content. Once the course content is supported with interactive animations, it will be more understandable and more striking. Presenting material in an animated manner has some advantages. Animations allow an instructor to emphasize the important parts of a subject by removing unnecessary and distracting details, and animations have the ability to visually present subjects and ideas that would be difficult to illustrate in the real world (Wetzel *et al.*, 1994). Flash animations are prepared to support the course content.

Fifth step is making the course SCORM compliant. The content should be compatible with Learning Management Systems developed by different vendors to reach more people and implement reusability. In order to make the content to work with LMS software developed by different vendors, SCORM is chosen, and the electronic content is organized as SCORM compliant.

1.2. Organization of the Thesis

This thesis consists of six sections:

- Section 1 is an introduction to the thesis. It makes the statement of the problem, and provides an overview of the thesis.
- Section 2 describes the software tools used to create the online course. Adobe Photoshop, Microsoft FrontPage and Macromedia Flash were used to create the electronic content.
- Section 3 gives information about LMS and SCORM. This section explains the importance of a SCORM compatible electronic content.

- Section 4 explains the structure and hierarchical design of the created web pages and navigation through the course content. This section also gives some examples of created animations and explains how to create these animations in detail. The chosen animations are just to give a general idea of how to create animations for this project.
- Section 5 documents the seven chapters in detail, with subchapters and created animations. Each subsection of Section 5 corresponds to one chapter. These subsections state the learning objectives for the chapter. Section 5 also gives information about how to make the course content SCORM compatible.
- Section 6 summarizes the project and makes recommendations for future work.

2. SOFTWARE TOOLS

There is a wide range of software tools that are available for the development of educational content. Each one has its advantages and disadvantages, which should be considered when evaluating the application one wants to develop. These tools are compared and suitable ones are chosen. This section gives information about the used software tools.

2.1. Adobe Photoshop

Adobe Photoshop is the most popular program for creating and modifying images for the WWW. This is true not only because Photoshop is available on a wide array of platforms ranging from Mac to Windows to UNIX, but because, Adobe Photoshop has the most intuitive user interface, the most complete set of tools, and the largest number of reference books around (Selena, 1998). There are other image editing tools, but for this project it is decided to focus on Photoshop primarily because Photoshop is the program that most web designers use.

Photoshop, now up to version 7.0, is the leading professional image-editing program, released by Adobe. Photoshop is useful for both creating and editing images to be used in print or online. Easy to use, but full of high-quality features, Photoshop is the best choice for any image manipulation job. Original images can be created as existing ones can be modified ones using a full collection of painting, drawing, and retouching tools of Adobe Photoshop (Adobe.com, 2004).

The visual quality of the created electronic content is one of the most important parts of this project. To achieve the visual quality, new images are created, or existing ones are formatted to fit to the course material. The following functions of Adobe Photoshop are mostly used to create and format images for the course content:

- Creating a new image
- Editing an existing image

- Cropping an image
- Resizing an image

Especially, cropping and resizing functions are used to format existing images to fit to the created Flash movies. Macromedia Flash can also resize images, but if the image is not resized with Adobe Photoshop and let Macromedia Flash to resize, it will result in bad quality Flash movies. Macromedia Flash is not as powerful as Adobe Photoshop by resizing existing images.

2.2. Microsoft FrontPage

Microsoft FrontPage is a tool for creating quick and impressive web sites. Texts can be created with FrontPage as though in a word processor. FrontPage will write all the html tags for you. If a graphic is needed, it is sufficient to drag it onto the page and FrontPage will convert it to .GIF or .JPG and write the html tags. Tables can also be created visually (Blue Star Learning, 2004).

For interaction, data sources like text files (.txt), HTML files (.htm) and even data files can be connected via ODBC. Built-in processing for forms and validation is included as well. FrontPage is also forward thinking with support for JavaScript, VBScript, Java, XML and dynamic HTML (DHTML).

FrontPage consist of two primary parts:

- FrontPage Explorer is a web creation and administration tool. It gives the web designer an overall picture of created web site. It also enables to build new web sites and later to change, maintain, administer, and publish the web sites.
- FrontPage Editor enables the designer to create and manage web pages. With the editor, text and paragraphs can be formatted, lists, tables, images can be inserted, and links can be created. Web pages can be viewed using the familiar What You See Is What You Get (WYSIWYG) interface similar to word-processing programs like Microsoft Word.

FrontPage 2002 has an integrated interface that lets the designer create and edit Web pages and manage entire Web sites, all within one application. All toolbars and menus are consistent with Microsoft Office applications and can be fully customized (Microsoft, 2001).

WWW is chosen to distribute the created electronic content. Each chapter is designed to deliver through web pages. To create the web pages without delay, Microsoft FrontPage 2002 is used as the web design tool. Basic editing tools like Microsoft Notepad and WordPad are also used in conjunction with Microsoft FrontPage to fix some minor errors on created web pages.

2.3. Macromedia Flash

2.3.1. What Is Macromedia Flash?

Macromedia Flash is a technology that allows animations, interactive forms, games and other lively features to be embedded in web pages. Macromedia Flash Player, which is required to play flash movies, is a well-known and trustworthy plug-in that users should feel comfortable installing. All new web browsers come this plug-in installed.

Macromedia Flash allows designers and developers to integrate video, text, audio, and graphics into immersive, rich experiences that deliver superior results for interactive marketing and presentations, e-learning, and application user interfaces. Flash is the world's most pervasive software platform, used by over one million professionals and reaching more than 97% of Internet-enabled desktops worldwide, as well as a wide range of devices (Macromedia, 2004).

Flash is the leading vector graphics technology for designing high-impact, low-bandwidth web sites. ShockWave Flash (SWF) is the file format used by Macromedia Flash to deliver graphics, animation and sound over the Internet. These techniques are rapidly changing the way we play, work, or just browse online (Reinhardt, 2001).

Vector graphics have certain characteristics that are due to how they are stored by a computer. A vector graphics file regenerates the image on the screen by using mathematics. For example, a circle includes information such as the location of the center, the radius, the line thickness, and the color. All the graphics created in Flash are vector based. Vector graphics have two advantages: The file size tends to remain small (therefore, it downloads fast), and the image can be scaled to any size without any degradation of the image quality (Kerman, 2003).

Additionally, SWF can deliver animation, rich colors, sound, and interactivity. Moreover, this approach downloads faster, it's scalable, and it boasts higher quality than other graphics formats. Vector graphics are ideal for the web because they are so lightweight (Mischook, 2004).

2.3.2. Why Choosing Flash?

HTML is ideal for creating static websites where text and images are placed at fixed positions. But it doesn't really support dynamic websites, where text, images, and animations are moving around on the screen. These effects were achieved with animated GIF images, Java Applets or Macromedia Flash.

In order to bring interactivity to the electronic content, animated GIF, Java Applets and Macromedia Flash are compared. Macromedia Flash is chosen to create interactive material in this thesis. Advantages of Flash against other technologies and the reasons for choosing Flash are stated below.

2.3.2.1. Flash versus Animated GIF. An animated GIF is actually many images saved in one. When the animated GIF is loaded onto a webpage, the browser simply loops the images. This means, if an animation of a clown that moves his hand up and down in 25 small movements is made, then the animated area of the image is saved 25 times. So it doesn't take a lot of animation to create a GIF image that's remarkable bigger than a regular clown image. Therefore, even small animations take forever to load. While animated GIFs can be used for animations, they do not support interactivity. They simply loop images in a predefined order (NetKontoret, 1999).

Compared to animated GIFs, the advantages of Flash are that:

- Flash movies load much faster, because they are smaller in size compared to animated GIFs.
- Flash movies allow interactivity, but animated GIFs does not support interactivity.
- Flash movies can use more than 256 colors, but animated GIFs are limited to 256 colors.

2.3.2.2. Flash versus Java. Java Applets are another approach to add dynamic effects to web pages. Java is a full-featured computer language. As such, it is enormously powerful, but also difficult. It is an excellent choice for many mathematical applications, but cannot match Flash for complete ease of creating animations and interactive Web pages (Cooper, 2002).

Java, the programming language for applets, requires programming skills. This means that the designer can't just start creating his own applets. Flash movies are in many ways similar to java applets. Small programs that can be embedded into his HTML pages. But unlike java applets, it is fairly easy to create animations in Flash with minor programming skills (NetKontoret, 1999).

Java typically does not require the user to pre-install a plug-in, but Flash plug-in can be downloaded very easily, and new generation web browser come plug-in already installed. Flash supports a wider range of graphic elements (GIFs, fonts, etc) and features (opacity, etc). Flash movies are smaller, more self-contained than Java, which result in faster downloads, faster loading (Drayman, 2002).

Other advantages of Flash are that, Flash movies are easier to create than java applets and Flash movies load faster than java applets. Lightweight vector graphics result in faster download from Internet.

2.3.3. ActionScript

ActionScript is the scripting language of Macromedia Flash. A scripting language is a way to communicate with a program; it can be used to tell Flash what to do and to ask Flash what is happening as a movie runs. This two-way communication results in creating interactive movies (Bleyle, 2001).

ActionScript is an object-oriented programming (OOP) language that is designed specifically for Web site animation. Originally released with Macromedia Flash 4 and enhanced for Flash 5, ActionScript is a sophisticated version of the script language introduced in Flash 3. ActionScript makes it possible for developers to create onscreen environments (such as games, tutorials, and e-commerce applications) that can respond to user input through the keyboard or mouse. ActionScript is an event-based language: just as is the case in real life, actions are triggered by events.

ActionScript was modeled on European Computer Manufacturers Association (ECMA) -262, an international standard for JavaScript. In the Flash 5 version, new ActionScript syntax, conventions, and features were introduced that make it similar to JavaScript, which in turn makes the language automatically familiar to most Web developers. Flash 5 also includes a new ActionScript editing environment that automates editing tasks and reduces development time (SearchWebServices.com, 2001).

In order to create interactive Flash movies, ActionScript programming language is used. Details about how to use it in movies will be discussed later.

3. LMS AND SCORM

3.1. Learning Management System (LMS)

3.1.1. What Is LMS?

A learning management system is an environment that enables communication, document sharing, content management, assessment and evaluation, and active learning among participants learning in a distance environment.

An LMS is used usually within an organization or a university in order to simplify the administration of the training/learning program and the distribution of the content. It enables administrators/teachers to deliver, track, analyze and report on their employees/students learning condition, schedule and catalogue the courses and offer assessments and report test scores. It also helps employees/students decide their learning path and communicate with their peers. It is often the case that LMSs provide editing software for creating Learning Objects but standalone creation tools such as Macromedia's Dreamweaver or Flash can also be used to create instructional content (Nichani, 2001).

3.1.2. Features of an LMS

An LMS provides a single point of access to disparate learning sources. It automates learning program administration and offers unprecedented opportunities for human resource development. It identifies the people who need a particular course and tells them how it fits into their overall career path, when it's available, how it's available (classroom, online, CD-ROM), if there are prerequisites, and when and how they can fulfill those prerequisites. Once learners complete a course, the LMS can administer tests based on proficiency requirements, report test results, and recommend next steps. In that capacity, LMSs are instrumental in assuring that organizations meet rigid certification requirements in such vertical markets as healthcare, finance, and government (Greenberg, 2002).

Look for these capabilities in an LMS:

- **Support for blended learning**: People learn in different ways. An LMS should offer a curriculum that mixes classroom and virtual courses easily. Combined, those features enable prescriptive and personalized training.
- **Administration tools:** The LMS must enable administrators to manage user registrations and profiles, define roles, set curricula, chart certification paths, assign tutors, author courses, manage content, and administer internal budgets, user payments, and charge backs. Administrators need complete access to the training database, enabling them to create standard and customized reports on individual and group performance. Reports should be scalable to include the entire workforce. The system should also be able to build schedules for learners, instructors, and classrooms. Most important, all features should be manageable using automated user-friendly interfaces.
- **Content integration:** It's important for an LMS to provide native support to a wide range of third-party courseware. When shopping for an LMS, keep in mind that some LMSs are compatible only with the supplier's own courseware, and others do little more than pay lip-service to learning content standards. An LMS supplier should be able to certify that third-party content will work within their system, and accessing courses should be as easy as using a drop-down menu.
- **Adherence to standards:** An LMS should attempt to support standards, such as **SCORM and AICC**. Support for standards means that the LMS can import and manage content and courseware that complies with standards regardless of the authoring system that produced it.
- **Assessment capabilities:** Evaluation, testing, and assessment engines help developers build a program that becomes more valuable over time. It's a good idea to have an assessment feature that enables authoring within the product and includes assessments as part of each course.
- **Skills management:** A skills management component enables organizations to measure training needs and identify improvement areas based on workers' collective competence in specified areas. Businesses also might use this feature to search their employee base for specialized skills.

As this thesis is concern, the important requirement of an LMS is SCORM support. If a SCORM compatible electronic content is prepared, this content should be followed on any vendor's LMS software.

3.2. Sharable Content Object Reference Model (SCORM)

3.2.1. What Is SCORM?

SCORM, which stands for "Shareable Content Object Reference Model", is a standard for developing, packaging and delivering high-quality training materials for online training courses. SCORM was developed based on the concept of "using a common e-learning standard to modernize education and training courses". Now, with the SCORM standard, and more specifically SCORM compliant tools/platforms, learning contents can found, imported, shared, reused and exported through compliant web-based learning systems (Cyberlink, 2002).

The Sharable Content Object Reference Model defines a Web-based learning "Content Aggregation Model" (CAM), "Run-Time Environment" (RTE) and sequencing and navigation for learning objects. The SCORM is a collection of specifications and standards adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reuse of Web-based learning content. The work of the Advanced Distributed Learning (ADL) Initiative to develop SCORM is also a process to bring together similar groups and interests. The ADL initiative was begun under the auspices of the U.S. Department of Defense in 1997 to create a Department wide strategy for using learning and information technologies to enhance education and training. A secondary goal is the development of e-learning standards by, for use by academia, government and industry. This reference model aims to coordinate emerging technologies and commercial and public implementations.

SCORM applies current technology developments to a specific content model by producing recommendations for consistent implementations. SCORM is built upon the work of the Aviation Industry Computer-Based Training (CBT) Committee (AICC), the IMS Global Learning Consortium, the Institute of Electrical and Electronics Engineers

(IEEE), the Alliance for Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) and others to create one unified "reference model" of interrelated technical specifications, standards and guidelines that meet Department of Defense (DoD) high-level requirements for Web-based learning content (Spigarelli, 2004).

3.2.2. Why Choosing SCORM?

Since the beginning of 2001, several notable learning resource specifications have gradually become mature after many years' efforts, which will provide with new opportunities to construct reusable and interoperable LMS and further exchange "standardized" learning resources between these LMSs.

There are three principal learning resource specifications: ADL SCORM, IMS Learning Resource Metadata Specification, and IEEE Learning Object Metadata (LOM). The SCORM smartly references IMS and LOM as well as other specifications and further integrates these specifications with one another to form a more complete and easier-to-implement model. With regard to the metadata definition, the SCORM directly references IMS Learning Resource Metadata Information Model, which itself is based on IEEE LOM. More importantly, the SCORM has defined its own Content Packaging Information Model that extends the IMS Content Packaging Information Model with several SCORM specific elements (Qu *et al*, 2001).

3.2.3. SCORM Components

SCORM defines a conceptual model on how to handle, package and deliver learning content. It encompasses a variety of learning standards (AICC, ARIADNE, IEEE, IMS and LTSC) in a model that aims to provide (Ghiglione, 2003):

- **Accessibility:** The ability to access training from remote locations and deliver those trainings in any locations.
- **Interoperability:** Training developed in any platform can be taken in any platform.
- **Durability:** Despite changes on the delivery technology, the learning resources should be still working fine (be delivered properly).

- **Reusability:** Training materials can be reuse in other contexts and/or courses.

Important parts of SCORM are,

Asset: Learning content in its most basic form - electronic media, text, images, sound, web pages, assessment objects or other pieces of data that can be delivered to a web client. Examples of assets are shown in Figure 3.1 (SCORM CAM, 2001). These assets are described with asset metadata to allow for search and discovery within online repositories, thereby enhancing opportunities for reuse.

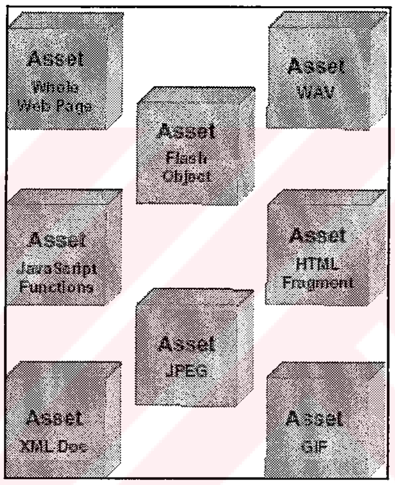

Figure 3.1. Asset (SCORM CAM, 2001)

Sharable Content Objects (SCO): An SCO represents a collection of one or more assets. It represents the lowest level of granularity of learning resources that can be tracked by an LMS using the SCORM run time environment. To be reusable it should be independent of learning context, so that it can be used in different learning experiences to fulfill different learning objectives. SCOs can also be aggregated to form a higher-level unit. An example of a SCO could be a web page consisting of HTML, GIF and/or JPEG images, WAV sound file, Flash Animation object, JavaScript function etc. An example of SCO is shown in Figure 3.2 (JCA Solutions, 2002). SCOs are described with SCO metadata to allow for search and discovery within online repositories, thereby enhancing opportunities for reuse. SCOs are bound to SCO metadata by the content package.

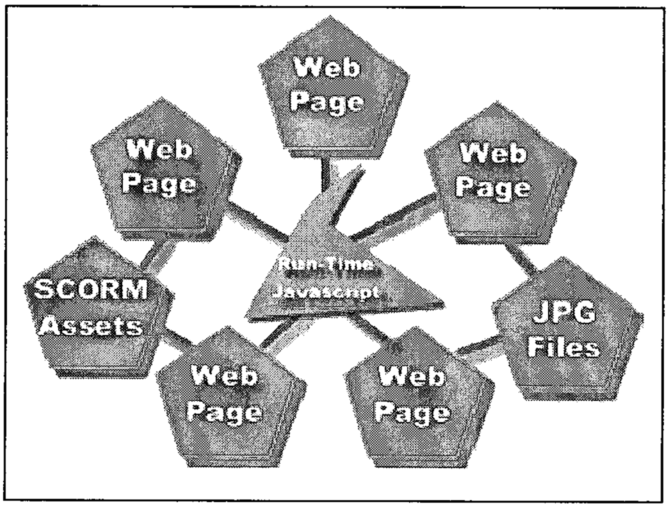

Figure 3.2. Example SCO (JCA Solutions, 2002)

Metadata: The purpose of meta-data (data about data) is to provide a common nomenclature enabling learning resources to be described in a common way. Meta-data can be collected in catalogs, as well as directly packaged with the learning resource it describes. Learning resources that are described with meta-data can be systematically searched for and retrieved for use and reuse (ADL, 2001).

3.2.4. Content Packaging

E-learning content is made up of a number of digital resources created by the content authors. These digital resources include things such as diagrams, text, animated pictures, audio or video clips, flash files, games, chat rooms, tests, or any other thing that can be created in digital media. To create instructional content, authors arrange the resources they create into coherent structure and sequence and package it. A Package is a collection of digital files that together completely contains a usable unit or units of instructional content. This might be:

- An entire course,
- Part of a course that has relevance outside of its course structure,
- A collection of courses or
- An unrelated collection of learning objects.

Packages are used as a common input/output format to move electronic learning content between LMS systems, authoring tools, content repositories or any other E-Learning system. The packaging specification describes data structures that are expressed in XML (NetDimensions, 2002).

SCORM 1.2 defines two kinds of Packages:

- **Content Aggregation Package** is a package that contains some structured unit or units of instructional content, and therefore, requires that at least one way of structuring the content be described in the organizations section of the manifest.
- **Resources Package** is a package used to transport collections of unrelated learning resources, and therefore, requires that the organizations section of the manifest be empty.

A Package consists of:

- A Manifest: This is a required XML document named 'imsmanifest.xml' that describes the resources (content files, video clips etc.) and structure (sequencing and navigation rules) of the content.
- All the physical files that are referenced in the manifest file, i.e. the actual html, Flash, audio/video or other content files that are described as resources in the manifest.

For easy distribution, all these files can be packaged into a single archive file using popular archive formats such as .zip, .cab or .tar. This archive file is called a Package Interchange File (PIF). The structure of a package is illustrated below in Figure 3.3 (NetDimensions, 2002).

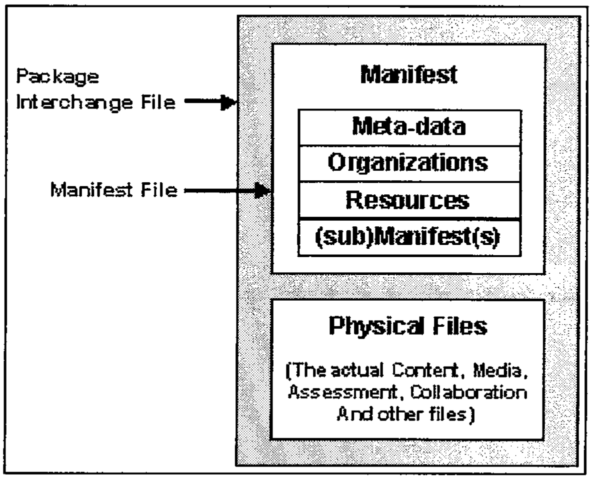

Figure 3.3. Structure of a packet (NetDimensions, 2002)

In this thesis, created electronic files are organized for SCORM standard to ensure that they will work on any LMS software. The files are defined in the manifest file named "imsmanifest.xml". With the information in this file, LMS can arrange prepared electronic content in a predefined structure.

4. STRUCTURE OF THE COURSE CONTENT

This section is about the structure of the course content. Hierarchical sequencing of the electronic files, navigation through the course content, and frame structure of the web pages will be explained. Some of the created animations are also explained to introduce the used animation methods.

4.1. Frame Structure

There are two types of frame pages:

i. For electronic content supported with animations,
ii. For electronic content not supported with animations.

Frame structure section explains these frame pages with examples.

4.1.1. Web Page for Animations – "index_2.html"

Web page named "index_2.html" is designed for texts which are supported by animations. The page is created with the frame feature of HTML. One example of these pages is shown in Figure 4.1.

This web page is divided in two rows. First row is divided in two columns. Left column is for texts and right column is for animations which support this text. Animations are created using Macromedia Flash. The second row is also divided into two columns. The left column allows navigating between chapters, and the right one allows navigating within the chapter. The important part of this design is the first row. There are two columns in this row, so this page is called "index_2.html", where "2" stands for the number of columns.

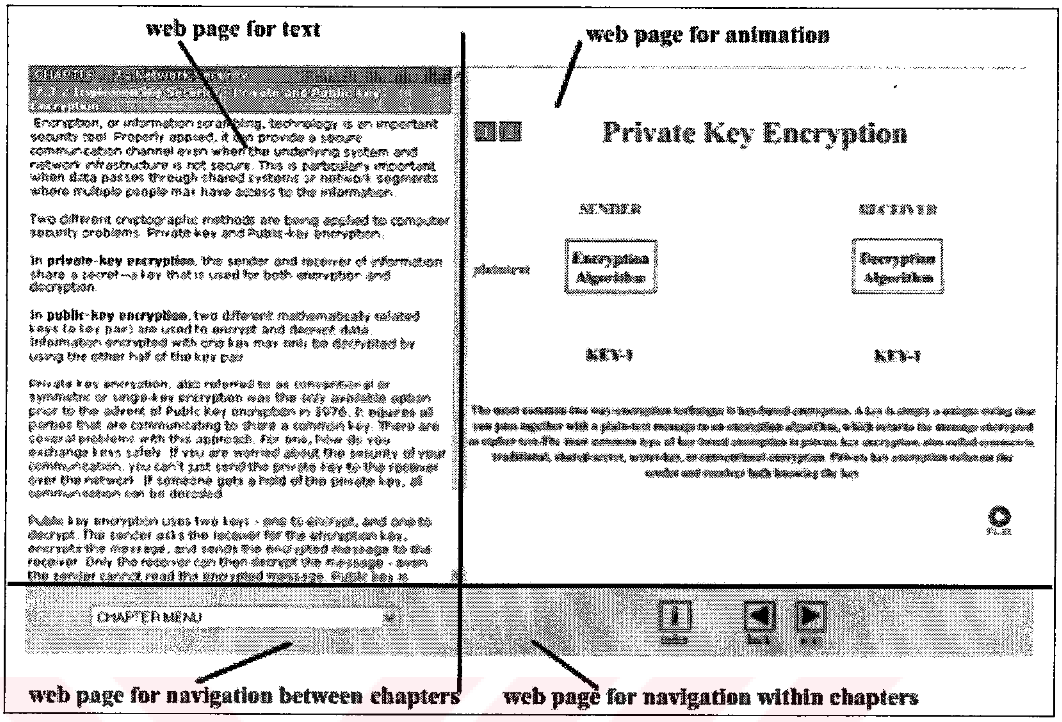

Figure 4.1. Web page design for animations

4.1.2. Web Page without Animations – "index_1.html"

Second web page design is for texts without animations. In this project, all of the texts are not supported with animations. These kinds of texts are submitted with another page design. This design is shown in Figure 4.2.

The difference between these two designs is only the first row. Web pages without animation have only one column in the first row, and this column contains the article. In this design, the page is called "index_1.html" where "1" stands for the number of column in the second row.

Actually, both of these pages are displayed in another frames page named "main.html", which will be displayed in section 4.2. Each chapter has one "main.html" which is called from another page named "start/index.html".

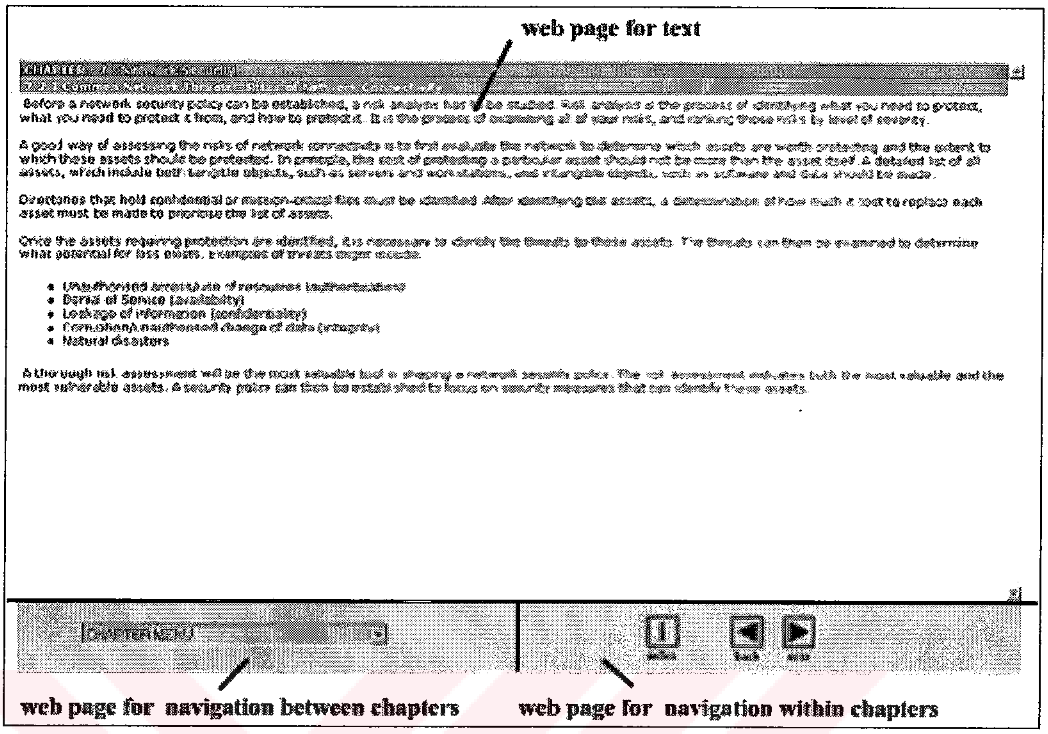

Figure 4.2. Web page design without animations

4.2. File Organization

This section explains the hierarchical structure of the electronic course files. This course is organized in seven chapters. Each chapter has topics, and each topic has subtopics. There is a three-level hierarchy.

Figure 4.3 shows the organizations of the electronic files. Figure 4.3 displays the seven chapters named “ch1” to “ch7”. In the figure, “ch1” is chosen and its four topics named “1_1” to “1_4” are shown. From these four topics, “1_2” is chosen and its six subtopics are displayed. Each subtopic has five required html files and one optional swf file for animations.

Two of these html pages, named “index_1.html” and “index_2.html” are explained in section 4.1. “button.html” is used for navigating within the chapter and calling table of contents (TOC). “content.html” is for text about the subtopic, and “media.html” is for animations which are used to support the text. If there is an animation file, the browser calls “index_2.html” to display the “media.html”, if there is no animation, the browser calls “index_1.html” which only displays “content.html”. The locations of these pages are shown in Figure 4.5. The file named “navigation.html” is located on the top level of this hierarchy and this file is used to navigate between chapters.

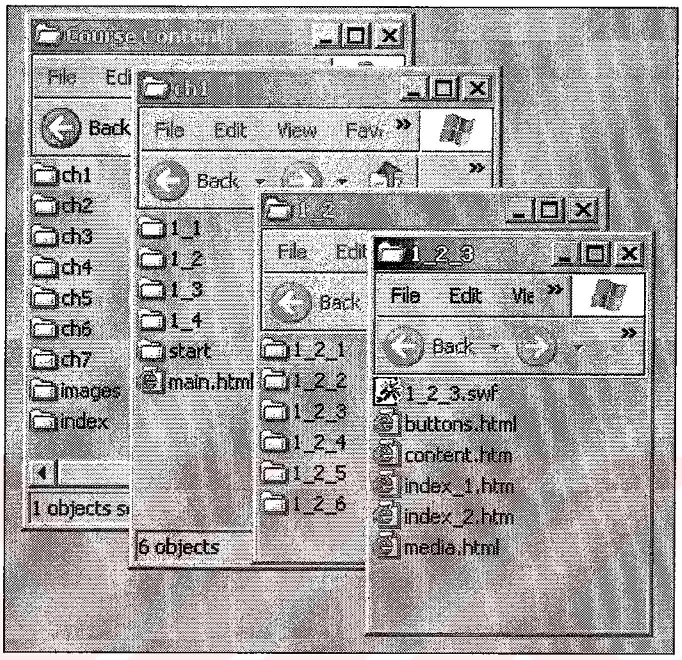

Figure 4.3. File organization of course files

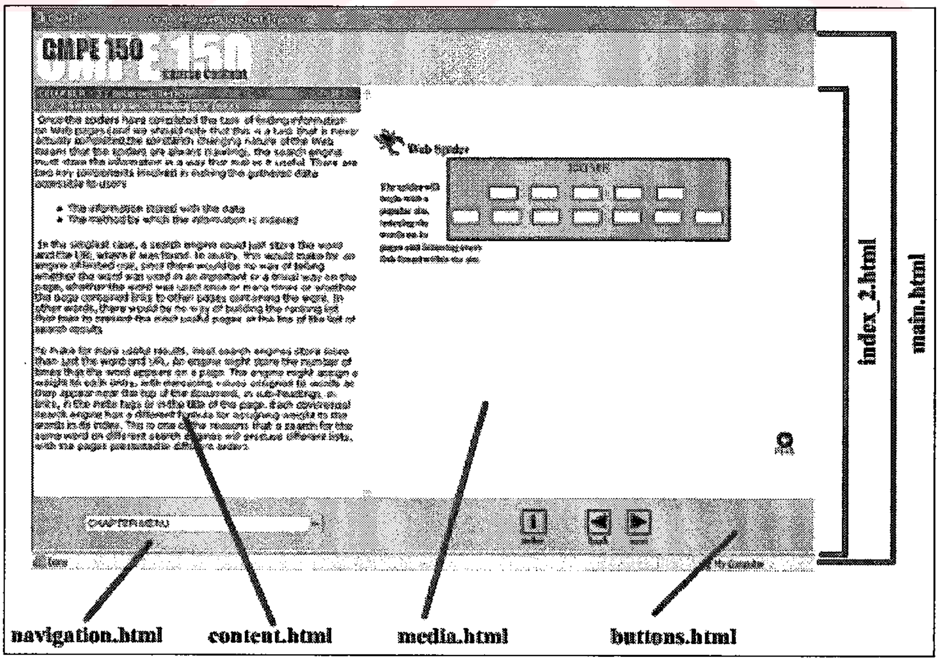

Figure 4.4. Page locations in frame page

4.3. Navigation through the Course Content

Electronic course content is prepared in HTML pages. To distribute the content, a user-friendly interface design is made. This section explains how to navigate through the course content. Navigation can be between chapters or within the same chapter.

4.3.1. Navigation using Home Page

The name of the home page is "start.html". Actually, another page, named "index.html" is called first, but "index.html" is just for checking the flash plug-in. If the computer does not have the required plug-in, "index.htm" calls a web page to download the plug-in.

Home page is shown in Figure 4.5. It is designed using flash. The names of the seven chapters are prepared as flash buttons, and the user can navigate using these buttons between seven chapters.

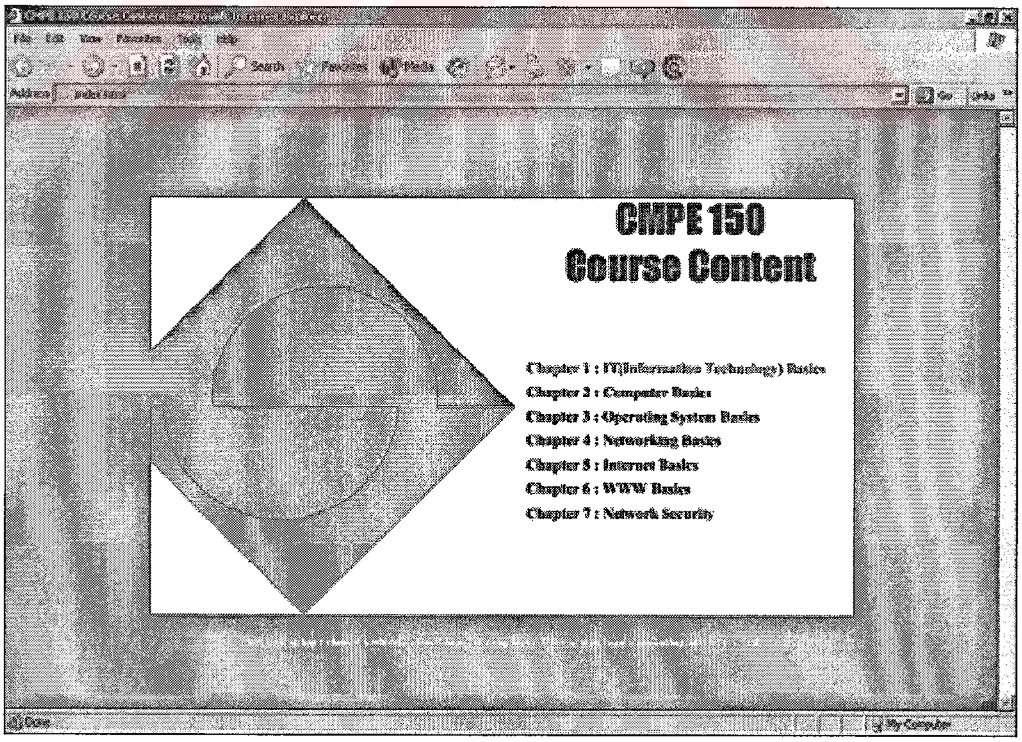

Figure 4.5. Home page

The ActionScript code written for button one is shown in Table 4.1.

Table 4.1. ActionScript code for button one

```
on (press) {
   getURL ("javascript:launchContent(1);");
}
```

This code calls "launchContent()" JavaScript function in start.html to display the content of Chapter 1. Code for "launchContent()" function is shown in Table 4.2.

Table 4.2. Code for "launchContent()" function

```
function launchContent(ch_num) {
      var httpRef1,httpRef2;


         httpRef1 = 'ch';
         httpRef2 = '/start/index.html';


      url = httpRef1+ch_num+ httpRef2;


      var newWindow = open(url,"currWin","width="+(screen.width-
10)+",height="+(screen.height-77)+",status=yes, resizable=yes, location=no, top=0,left=0,
scrollbars=yes, toolbar=no, menubar=no");
   }
```

Choosing one of the seven buttons in home page lets the user to open the selected chapter content. The function in Table 4.2 takes selected chapter number (ch_num) and opens this chapter. Using home page, the user can navigate between chapters.

4.3.2. Navigation using "navigation.html"

An option to navigate between chapters is using "navigation.html". This page has a drop-down menu to choose one of the seven chapters. Figure 4.5 shows what happens when the user click on the chapter menu in navigation.html. Required chapter can be selected using this menu.

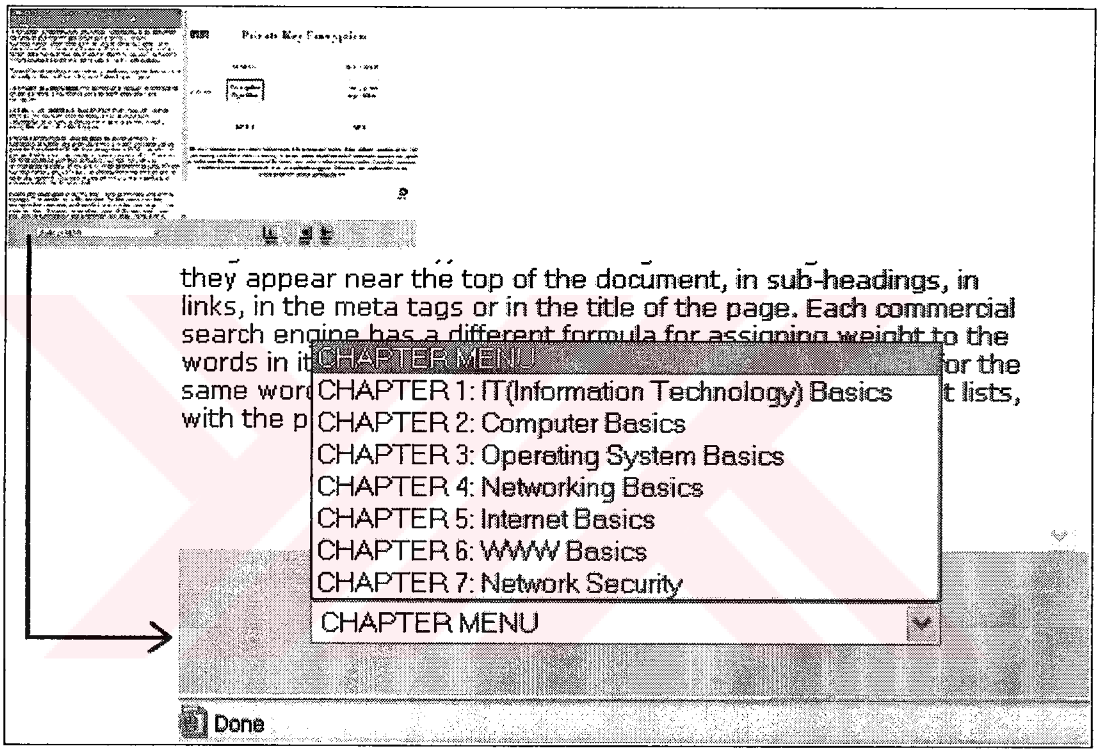

Figure 4.6. Navigation page

4.3.3. Navigation using "buttons.html"

Using "buttons.html" is another option to navigate the course content. Location of buttons.html is displayed in Figure 4.4. This page has three buttons. One of them is used to launch table of contents. Table of contents will be explained in the next section. Other buttons are "back" for navigating to previous page, and "next" for navigating to the next page. In short, "back" and "next" buttons are used to navigate within the chapter, "index" button is used to open table of contents which permits to navigate between chapters or within the same chapter. Figure 4.7 shows "buttons.html".

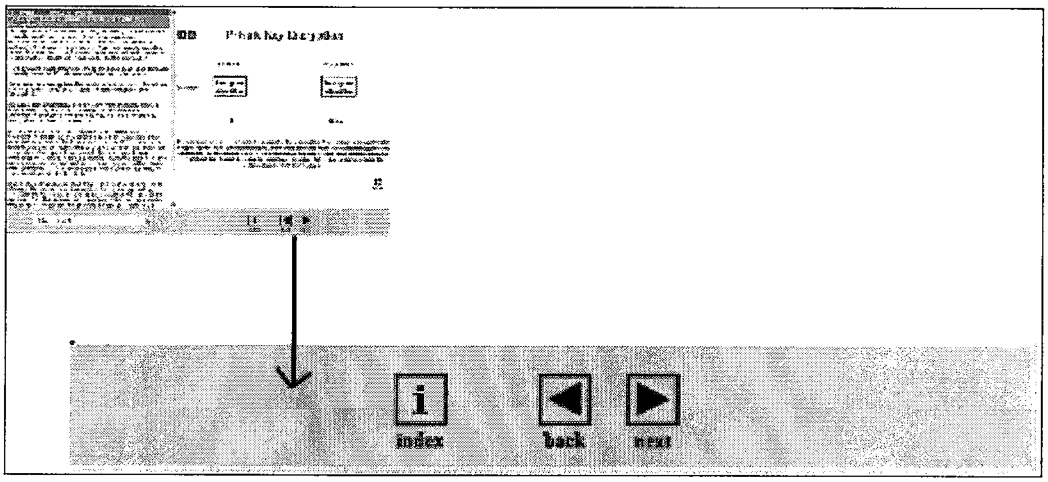

Figure 4.7. Buttons page

4.3.4. Navigation using Table of Contents

The last option to navigate in the course content is using TOC. TOC can be accessed using “index” button on the “buttons.html” page.

TOC opens in a new browser window, and the user can choose the subject from this course index page. The JavaScript functions to open the TOC are in Table 4.3.

Table 4.3. Functions to open the TOC

```
var w = null;
var c = null;

function getChapter()
 {
   if(!w)
     w = window.open('','indexWin');
   w.frames[1].location = "ch" + c + "/index.html";
   w.focus();
```

```
  }


 function openIndex(x,k)

  {

    c = x;

    w = window.open('../' + k +
'/index/outline/index.html','indexWin','toolbar=no,menubar=no,location=no,
scrollbars=yes,height=600,width=490,left=1,top=1,resizable=yes');


  }
```

Each chapter has its own course index. Parameter "x" is for chapter number. Once the TOC of one of the seven chapters is chosen, other chapter's course index pages are also accessible. Figure 4.8 shows the TOC and the functions of its links.

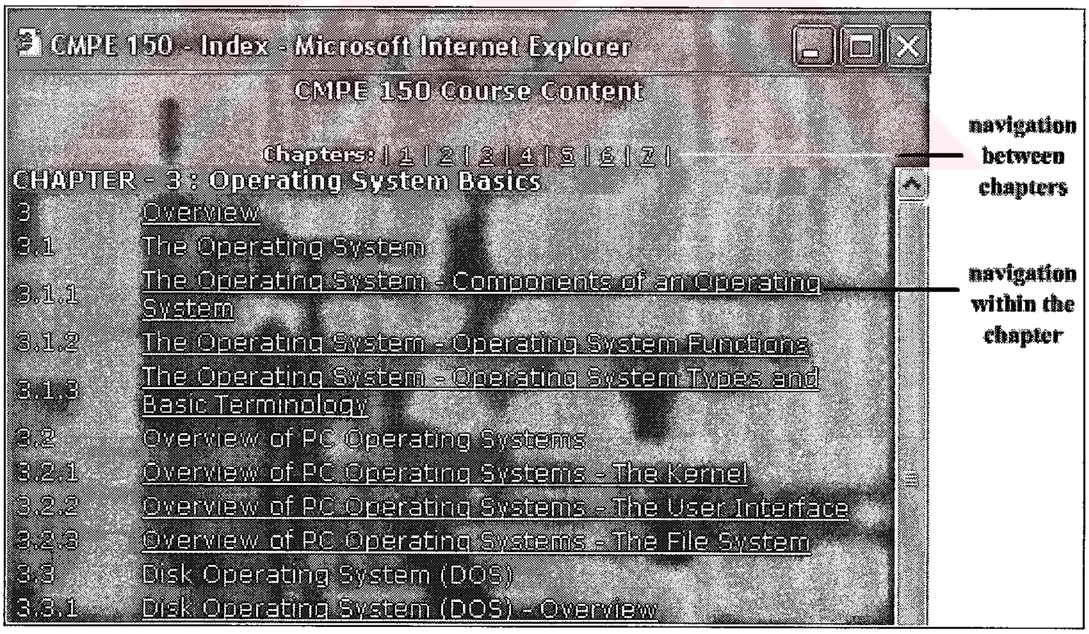

Figure 4.8. TOC

4.4. Examples of Animations

In this section, some of the created animations are selected to explain in detail. These selected animations can give a general sense of how the animations are created in this project. They are just some examples to explain the technique to create animations; other animations are explained in section five.

Selected animations and their chapter numbers are:

- Chapter 1: Quiz
- Chapter 2: 2.2.5 Hardware Components - The CPU
- Chapter 4: 4.3.6 OSI Reference Model - The Functions of Each Layer
- Chapter 4: 4.4.5 TCP/IP Model - Transmission Control Protocol
- Chapter 5: 5.3.2 WWW - Web Browsers
- Chapter 5: 5.4.2 E-MAIL - Address Structure
- Chapter 5: 5.7.2 Search Engine - Building the Index
- Chapter 4: 4.2.1 General Communication Terms - Source, Destination, and Data Packets
- Chapter 7: 7.3.2 Implementing Security - Private and Public Key Encryption
- Chapter 7: 7.3.5 Implementing Security - Firewall

4.4.1. Quiz

Each chapter has a quiz at the end. Flash animation prepared for quiz pages is explained in this section. Figure 4.9 shows quiz page of chapter one, IT Basics.

There are five questions and two buttons for each question. Number buttons are for questions, and "A" buttons are for answers. If one of the question number is pointed, a small red arrow appears beside the pointed number, and selected question is displayed on the screen.

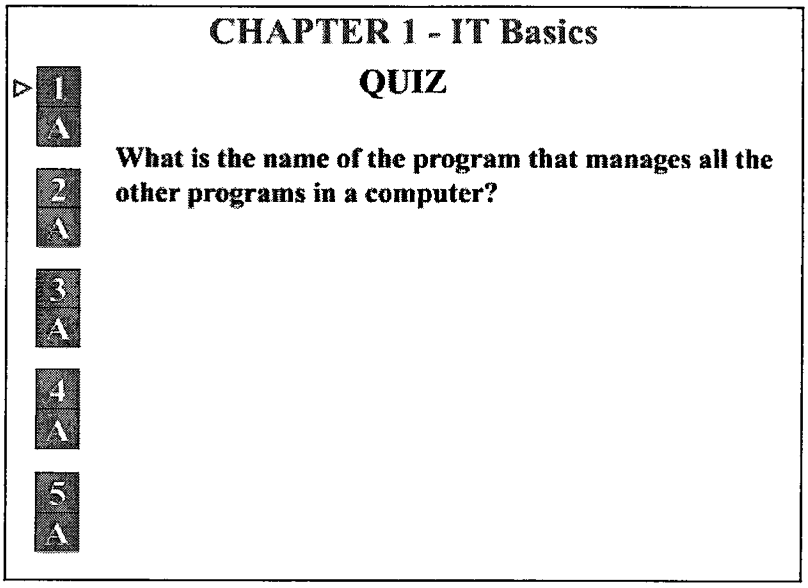

Figure 4.9. Quiz - question

If one of the answers is pointed, a small red arrow appears beside the pointed "A" button, and selected answer is displayed on the screen. Figure 4.10 shows an example where an answer is chosen. With the red arrow, the user can trace which question, or answer is displayed on the screen.

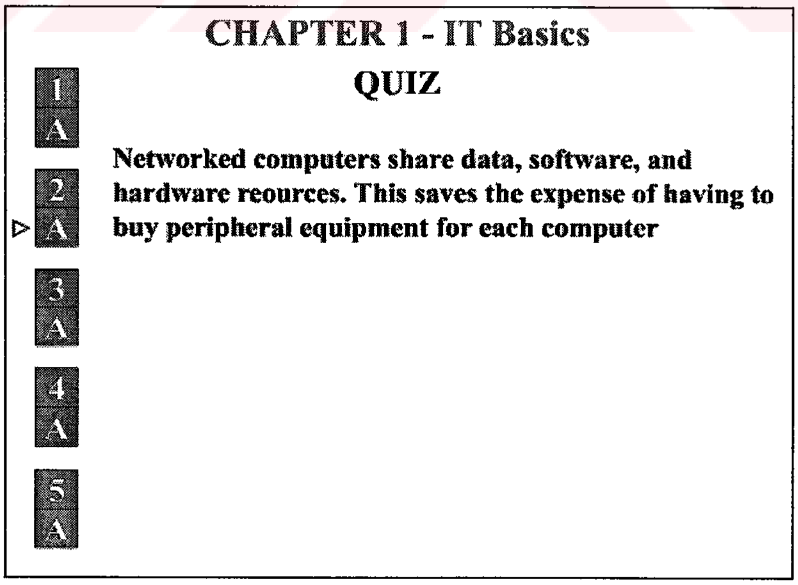

Figure 4.10. Quiz - answer

This flash movie is prepared using buttons and ActionScript. The question numbers and answers are buttons. If the user points the mouse one of these buttons, the number, or the letter on the button turn black from white, and executes the ActionScript code written for this button. Flash editor view to create this movie is shown is Figure 4.11.

Each question and each answer is prepared on a separate frame. Fixed parts, such as the buttons are put in a layer named "buttons", non-fixed parts, such as questions, answers and the little red arrow are put in the layer named "frames". Layer "frames" has ten different frames for these five questions and answers, and on each frame, another question or answer is displayed. If, for example, button number three is chosen, the movie displays frame number five and stops at this frame. ActionScript code written for this button is shown in Table 4.4. This code tells to go to frame number five and stop.

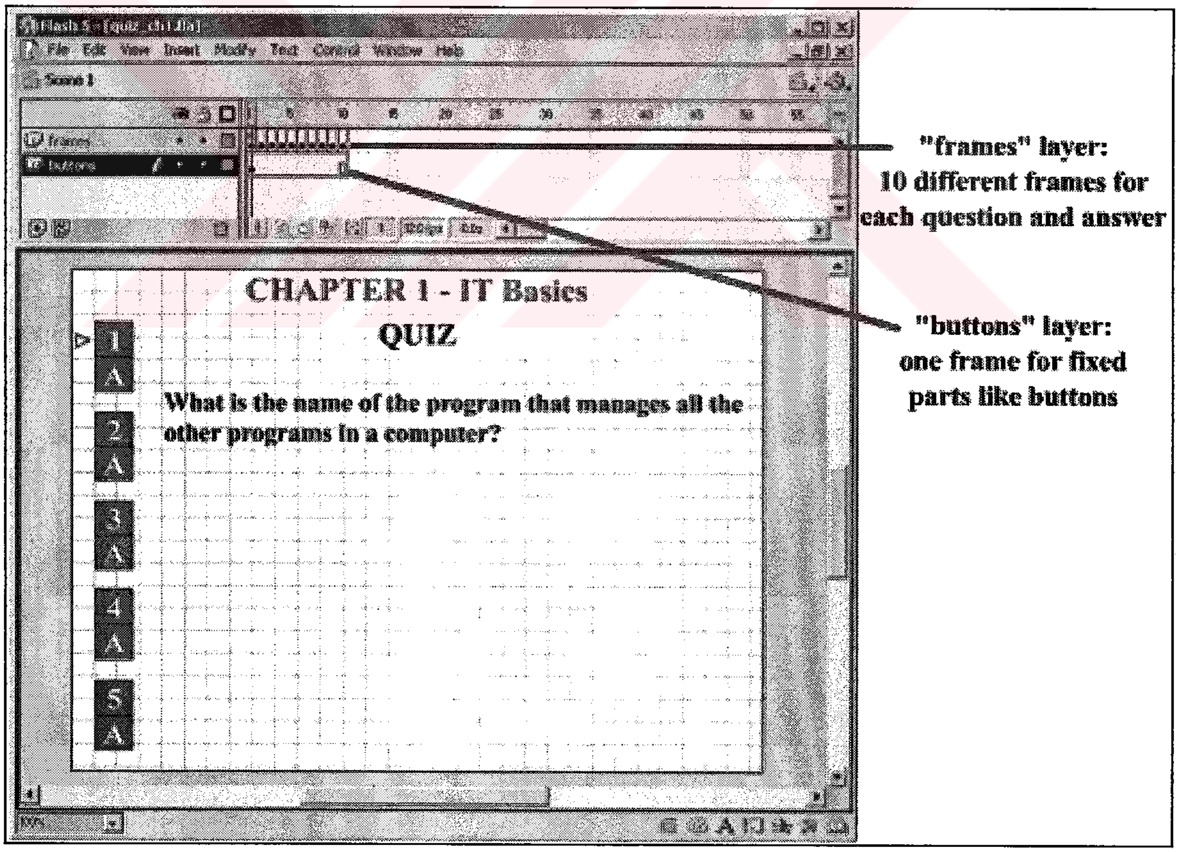

Figure 4.11. Quiz – editor view

Table 4.4. ActionScript code for button three

```
on (rollOver)
{
   gotoAndStop (5);
}
```

4.4.2. The CPU

Second animation is selected from Chapter 2. This animation is about the CPU. The user can view chief CPU types using this animation. Figure 4.12 shows this animation.

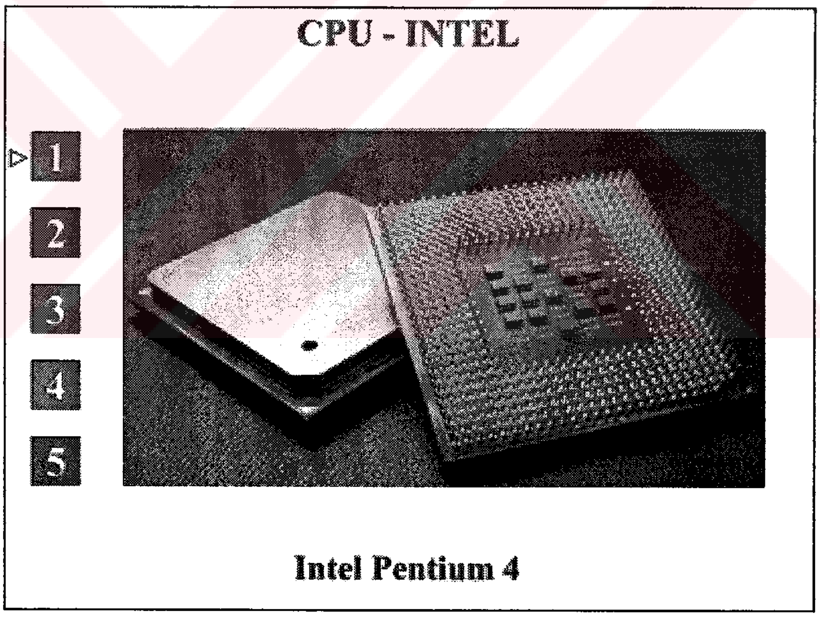

Figure 4.12. The CPU – button 1

The user can select buttons from one to five. Each button displays another CPU type. Figure 4.13 shows what happens if button three is selected. Displayed CPU type is changed. The red arrow changes its location and appears beside the button number three. This feature lets the user to examine five different CPU types by pointing the buttons.

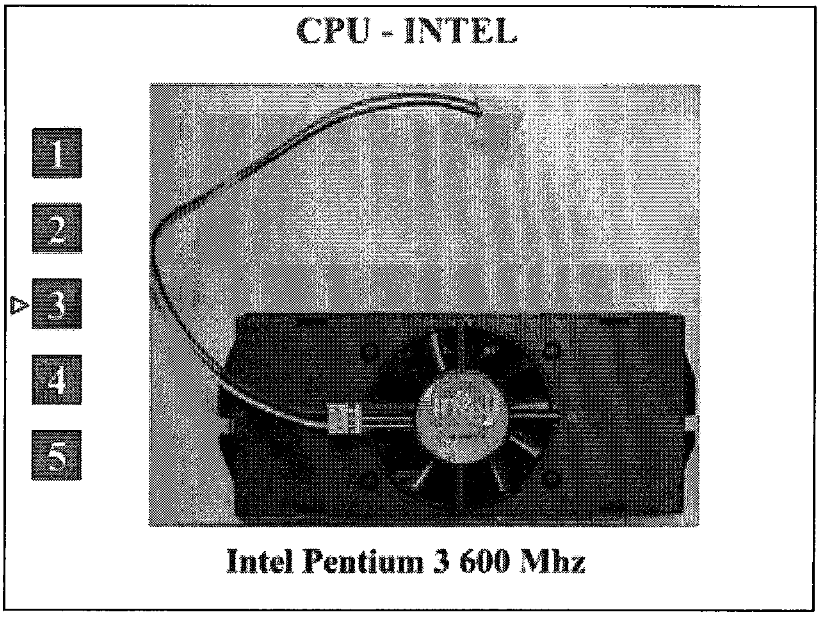

Figure 4.13. The CPU – button 3

Flash editor view is shown in Figure 4.14. The design concept of this animation is similar to "Chapter 1: Quiz" animation. Five buttons are created and each button is given a task using ActionScript.

There are two layers in flash movie. One is for fixed parts, named "Buttons" layer; other one is, named "Settings" layer, for non-fixed parts, like CPU images, CPU description and red arrow, which points the selected button. For fixed parts, creating one frame and copying this frame to other frames is enough. For non-fixed parts, a new frame should be created. These new frames are called "keyframe" in flash. A keyframe is a frame where changes in the animation are defined. When frame-by-frame animation is created, every frame is a keyframe.

In this animation, five different keyframe is created in "Settings" layer. Each keyframe has another CPU image. The user can navigate between these keyframes using buttons. These buttons are instructed to display selected keyframe by using ActionScript.

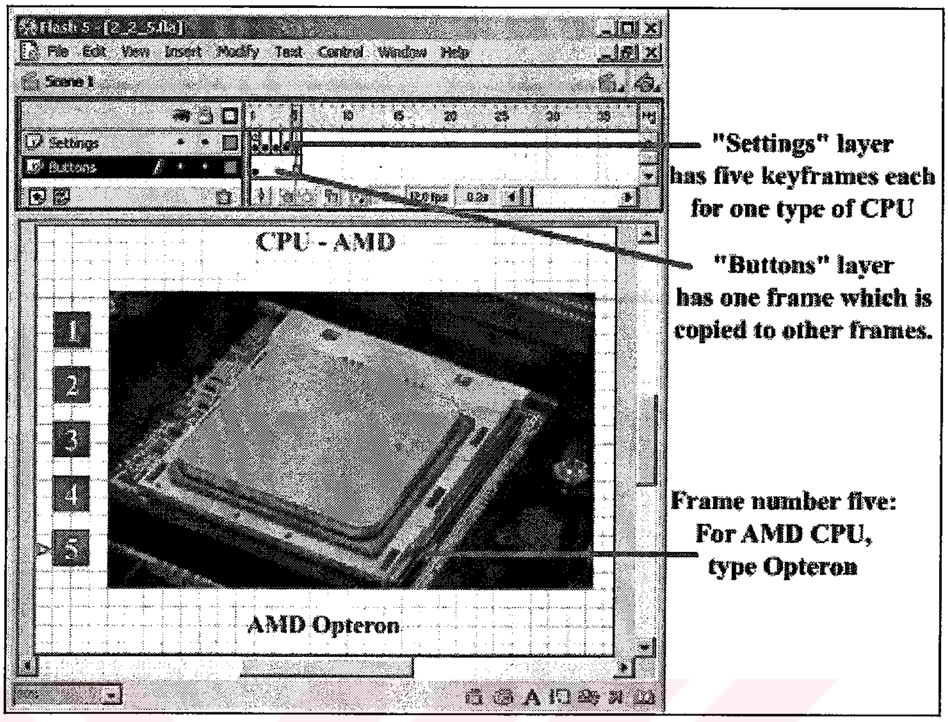

Figure 4.14. The CPU – editor view

Electronic content is created for a screen resolution of 1024 x 768. At this resolution, available place for animations is 576 x 431. Flash movies are created at this resolution, and the images in the movies also should be prepared to fit this area. In order to resize these images, first, Macromedia Flash's resize tool was used, but results are not good enough. Then, Adobe Photoshop is used to resize these images.

4.4.3. The Functions of Each Layer

Third animation is taken from chapter four. Figure 4.15 shows this animation. This animation is about the functions of each OSI layer. The user can view seven OSI layers and their functions using this animation.

When the user points one of the seven layer names, a comment about the corresponding layer appears on the right side. If the user points one of the layer names, pointed layer name turns blue while mouse pointer is on this layer, and the comment also changes to the pointed layer. Figure 4.16 shows what happens when the user points "Transport" layer.

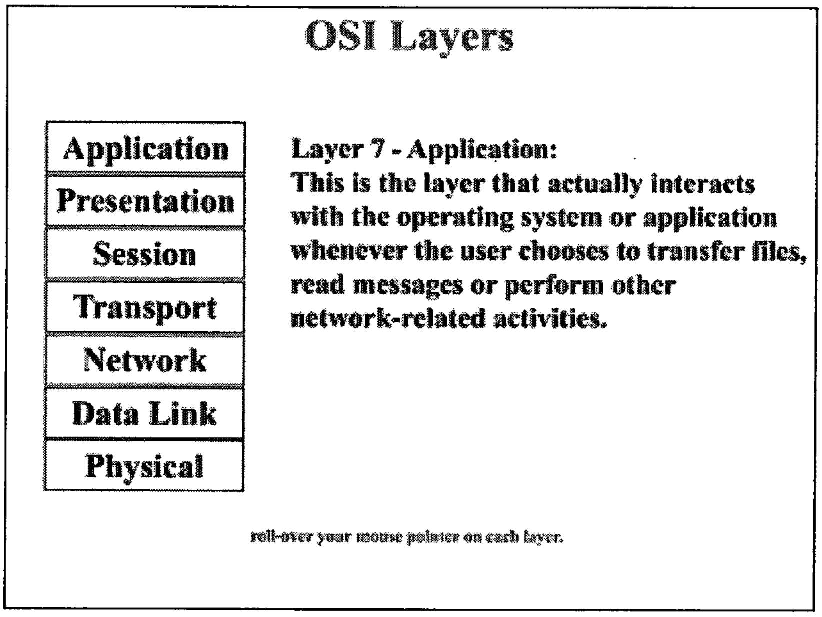

Figure 4.15. The functions of each layer - application layer

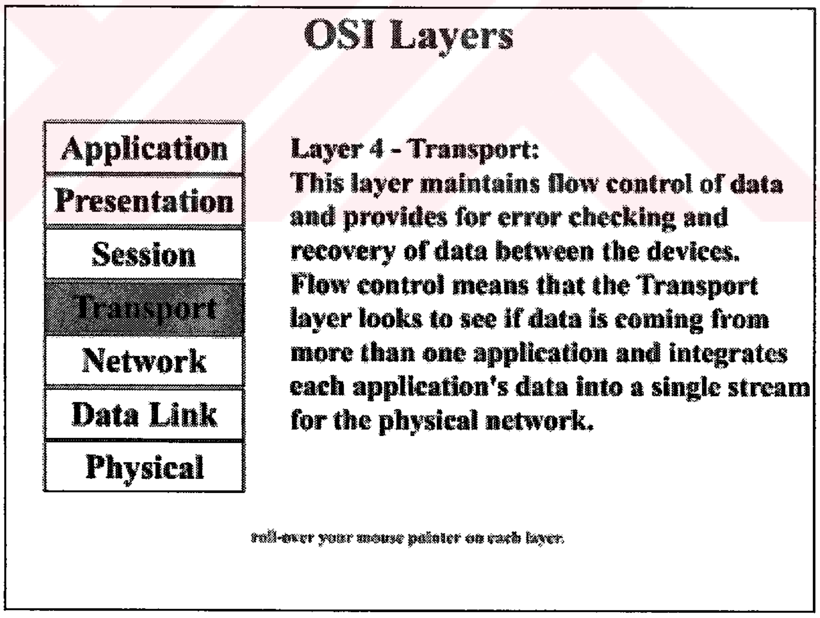

Figure 4.16. The functions of each layer - transport layer

In this animation, each layer name is designed as a button. Comments about the seven OSI Layers are defined in seven keyframes, one for each layer. The buttons are given tasks to display corresponding keyframe as the user points one of them. To warn the users about the buttons, there is a message in red font, which says "roll-over your mouse pointer on each layer".

Each button can have three different behaviors. These are:

- Up: Normal behavior.
- Over: When mouse pointer is over the button.
- Down: When user clicks the button.

To create a button, images for the up, over, and down states should be created. For this animation, seven different buttons are prepared. In a flash animation, buttons can be selected by using Window/ Common Libraries/ Buttons link. These are predefined buttons of Macromedia Flash. They can be used directly, or they can also be changed for the purpose. If a new button design is required, like in this animation, Insert/ New Symbol/ Button can be chosen. To edit a button, right click on it and choose "Edit" from the menu. The editor to edit a button is shown in Figure 4.17. The three states of a button are displayed in the figure. "Over" state is chosen. It means, when the mouse pointer comes over this button, the color of the button background turns blue and the color of the word turns black.

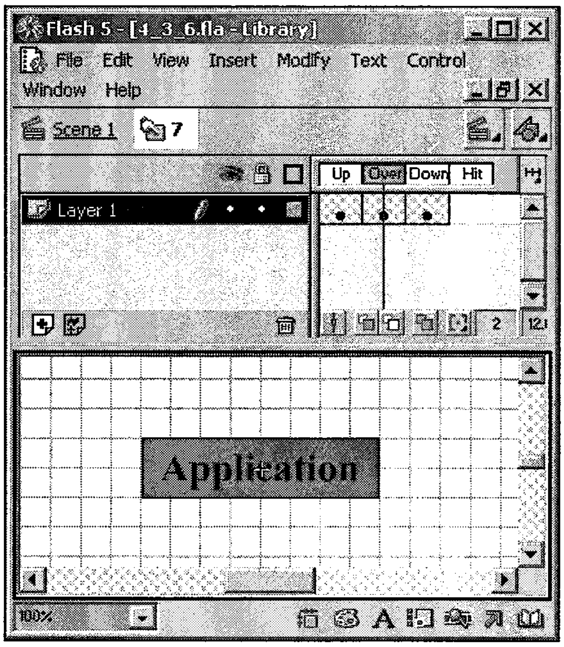

Figure 4.17. The functions of each layer - editing a button

4.4.4. Transmission Control Protocol

Fourth animation is selected from chapter four, Networking Basics. This animation is about the three-way handshake in TCP/IP. Animation is shown in Figure 4.18. There are two computers, one is server, and other is client. The user can let this flash movie play by clicking on the play button. An ActionScript code is written for this button. This code shows the frames consecutively until a frame with stop() action is arrived.

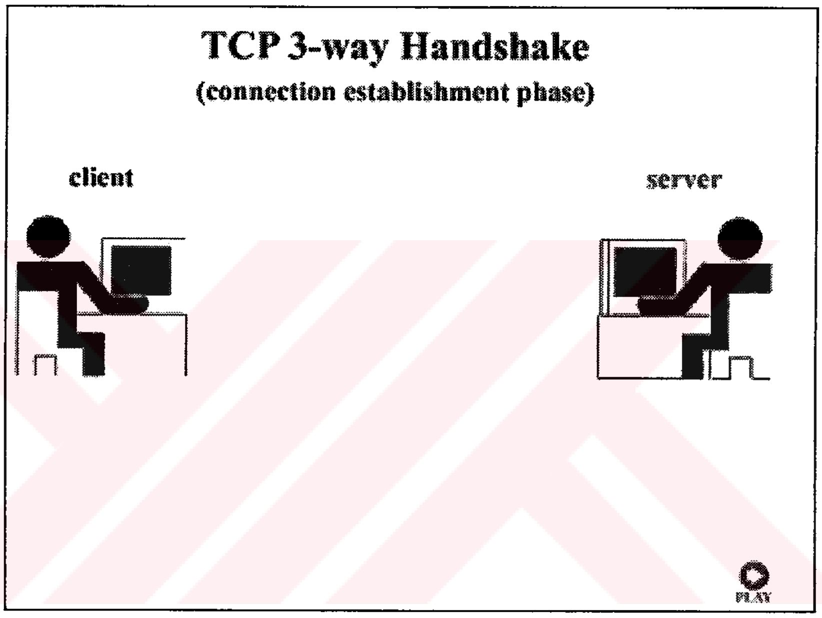

Figure 4.18. Transmission control protocol - start of the animation

Figure 4.19 shows what happens when a frame with stop() action is arrived. This is the end of the animation. Another button, named “back” is appearing on the same location of “play” button. By clicking on this button, the movie goes back to frame number one, and can be played again.

In this animation, another feature of flash, named “tweening” is used. This animation consists of three layers. First one is for the fixed parts, like the computers and the name of the animation. Second layer is for non-fixed parts, the arrows between the computers and comments about the arrows, and the third layer is for play and back buttons.

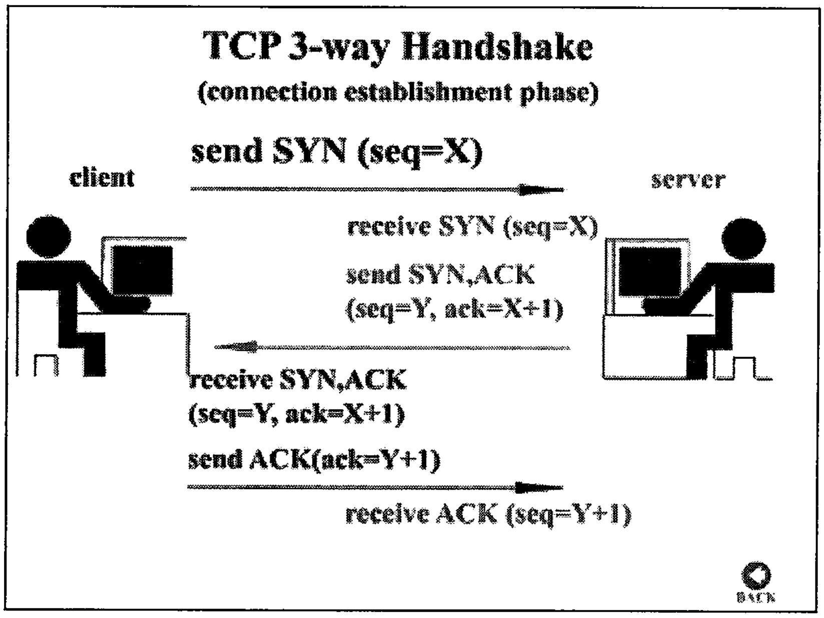

Figure 4.19. Transmission control protocol - end of the animation

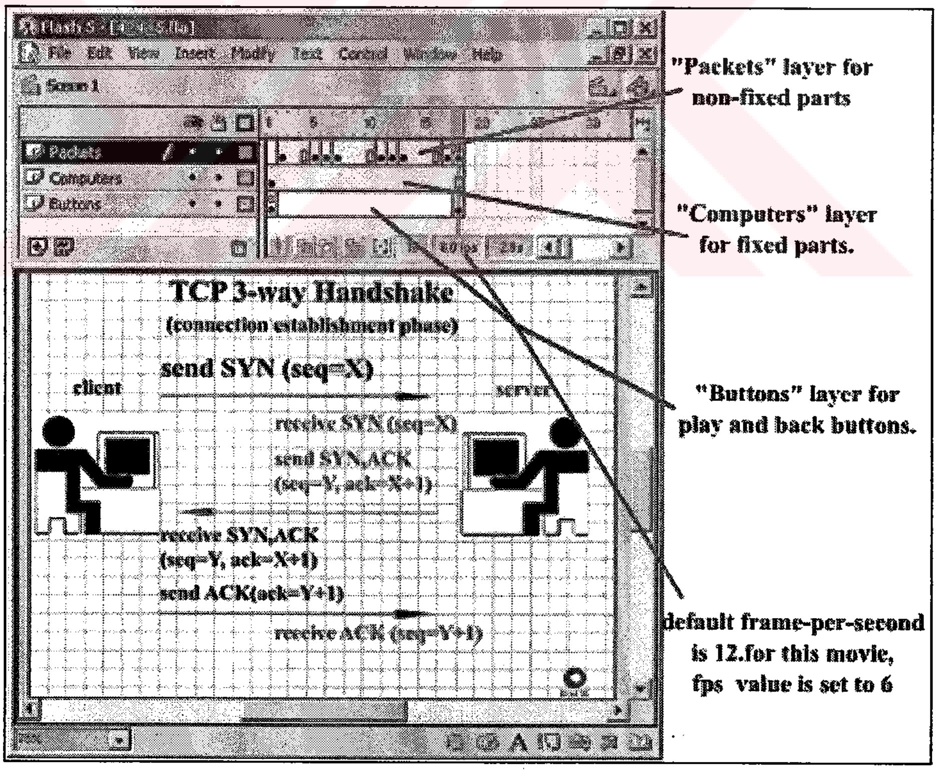

Figure 4.20. Transmission control protocol – editor view

Figure 4.20 shows editor view of this animation. This view is taken from the last frame of the animation. FPS value of this animation is set to six. Default value is twelve. There is a stop() action on the first frame. If the user clicks on the play button, the animation plays until another stop() action is met. Other stop() action is on the last frame of the animation

There is another button on the last frame, named "back". This button has the ActionScript code in Table 4.5.

Table 4.5. ActionScript code for back button

```
on (release)
{
   gotoAndStop (1);
}
```

This code tells the movie to go to frame number one and stop. This means, if the user clicks on the back button, the animation begins again.

In this animation, arrows represent the messages between client and server. These arrows are getting bigger during the movie plays. This effect can be achieved using the tweening feature of flash. For shape tweening, the first view and the last view of the arrow are created. Frames between these two are done by flash. While the movie plays, the arrow seems getting longer.

4.4.5. Web Browsers

The fifth animation is about the web browsers. This animation is taken from chapter five, Internet Basics. In this animation, the tasks of the buttons on a web browser are explained. Figure 4.21 shows the animation. The buttons of a web browser are formed as flash buttons.

If the user points one of the browser buttons, the name of the button and a comment about this button is displayed below the line. A red arrow appears beside the pointed browser part. Figure 4.22 shows what happens when another browser button is chosen.

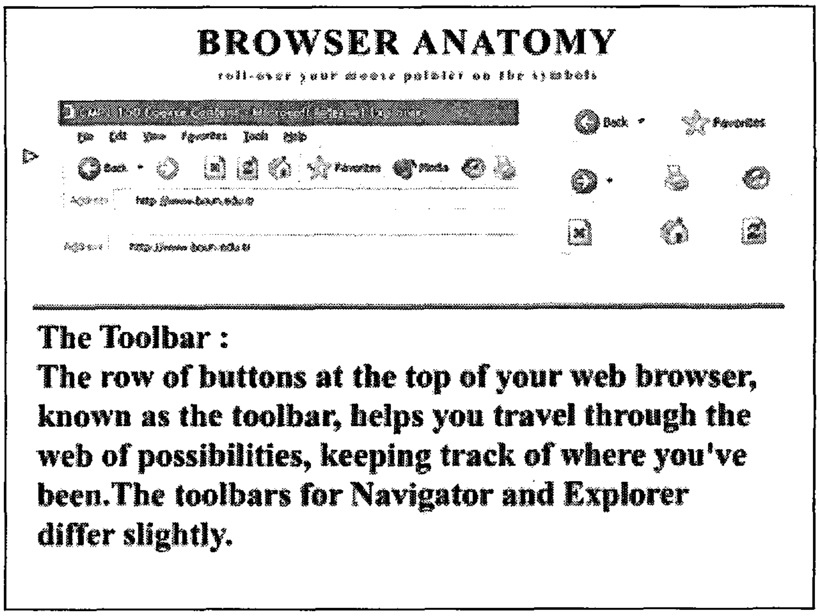

Figure 4.21. Web browsers - the toolbar

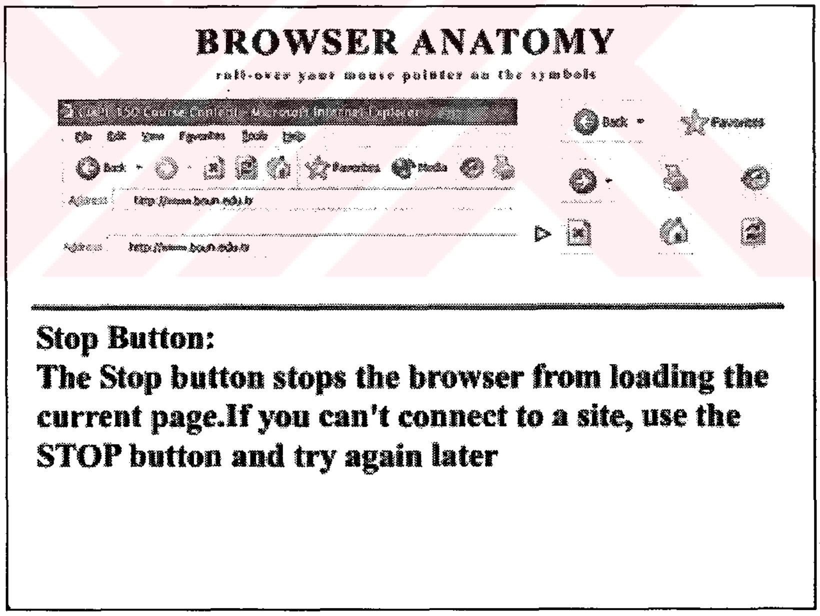

Figure 4.22. Web browsers - stop button

This time, the stop button is pointed with mouse and comment about the stop button is displayed. The red arrow appears beside the stop button.

In this animation, each button of a web browser is designed as a flash button. A browser image is taken and the required parts are sliced using Adobe Photoshop slice tool.

These sliced parts are converted to flash buttons. Then, for each flash button, an ActionScript code is written. Editor view of this animation is in Figure 4.23.

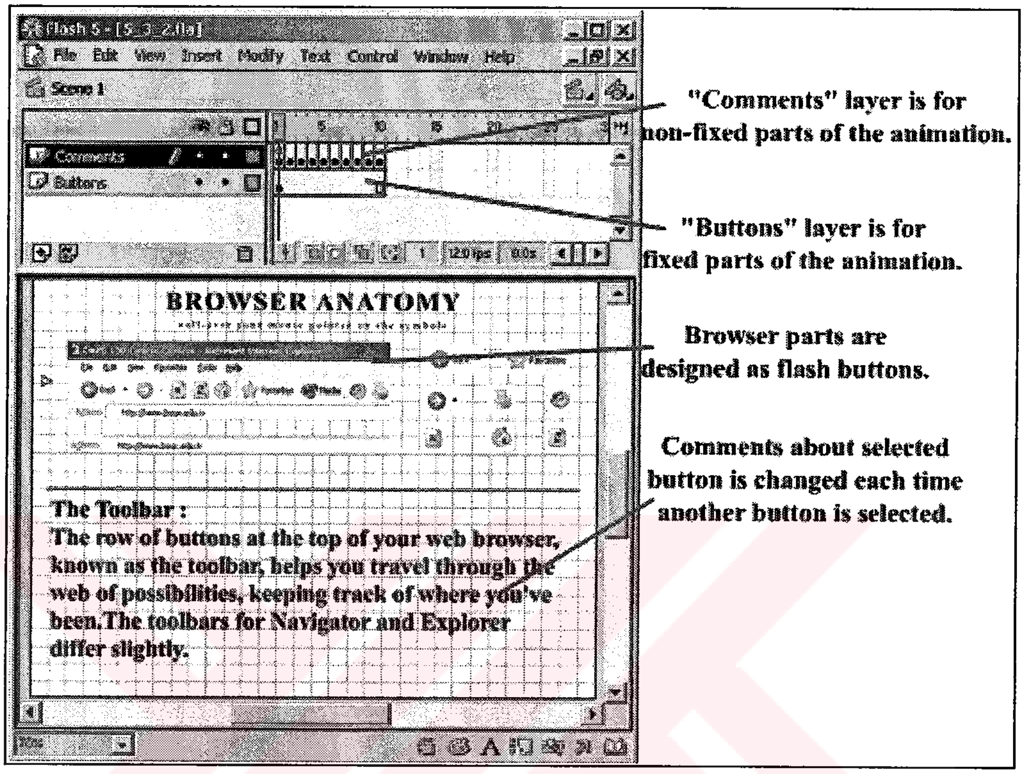

Figure 4.23. Web browsers – editor view

There are two layers in the movie. "Comments" layer is for non-fixed parts, like location of red arrow and comments about selected web browser button. They change for every selected button. "Buttons" layer is for fixed parts, like location of the browser buttons and the name of the animation.

In "Comments" layer, ten different frames are constructed for each browser button. In these frames, button name and description of the button are given. When one of the flash buttons in the "Buttons" layer is selected, ActionScript tells this button to play the appropriate frame; and this frame displays the comments written for the selected button.

4.4.6. Address Structure

The sixth animation is about the e-mail and web page address structure. This animation is taken from chapter five, Internet Basics. In this animation, the parts of an e-mail address and a web address are explained. Figure 4.24 shows the animation.

Figure 4.24. Address structure - username

Each part of e-mail and web addresses is constructed as flash buttons. If the user points one of the address parts, a red arrow appears under the pointed part and a comment about the related come into view.

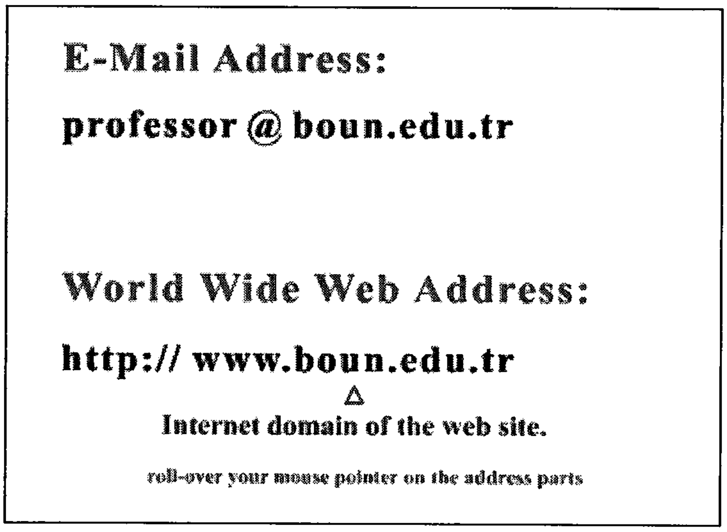

Figure 4.25. Address structure - name of the web site

Figure 4.25 shows another view of this animation. This time, Internet domain name part of the web site address is pointed. Red arrow appears under the domain name, and the description of the pointed part come into view under the red arrow.

In this animation, each part of these addresses is designed as a flash button. The whole address is written, and then address parts are converted to flash buttons. Each button is given a task using ActionScript which is to display a comment about the pointed address part. Figure 4.26 shows editor view of the animation.

There are two layers in the movie. “Comments” layer is for non-fixed parts, like location of red arrow and information about selected part of the e-mail address or web address. “Buttons” layer is for fixed parts, like location of the address part buttons. If one of the address parts is pointed, the red arrow appears under the address part and appropriate frame is displayed. Each frame in the “Comments” layer is a comment about an address part. ActionScript instructs the buttons which frame to display. To convert a written word to a button, choose the word, then select “Insert/ Convert to Symbol” from flash menu, and choose “Button”.

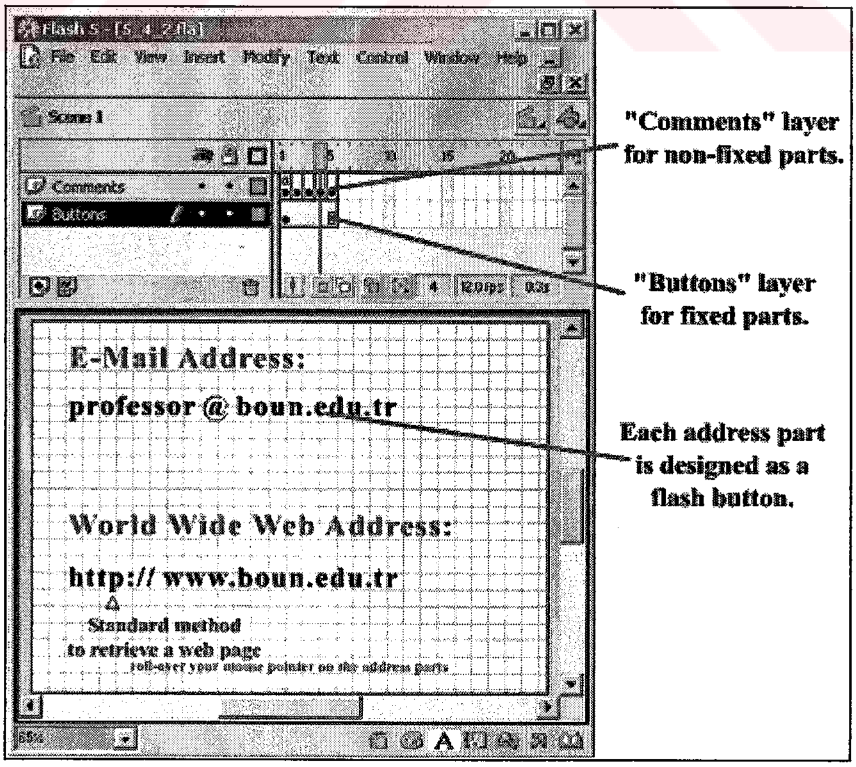

Figure 4.26. Address structure - editor view

4.4.7. Building the Index

The seventh animation is about the search engines. This animation is taken from chapter five, Internet Basics. In this animation, building the index of a search engine is explained. Figure 4.27 shows this animation.

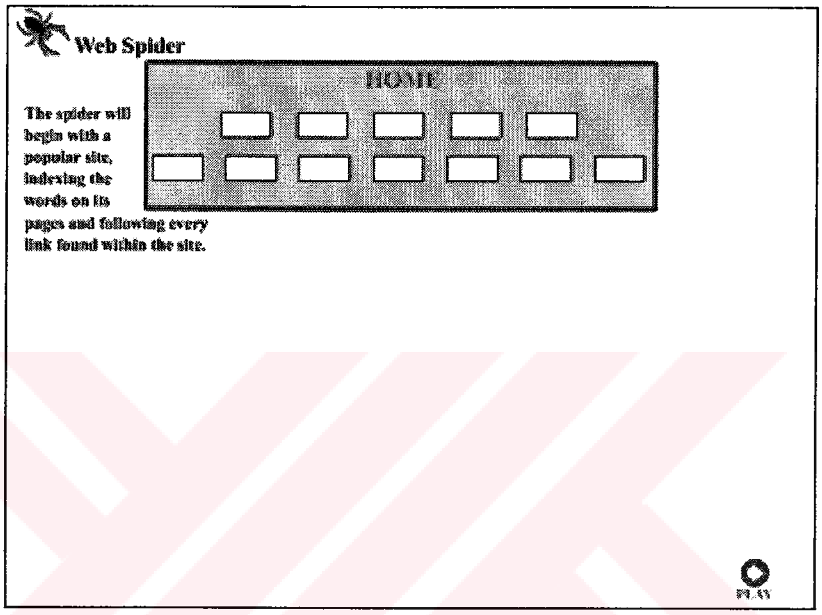

Figure 4.27. Building the index - start of the animation

This animation begins after pressing "play" button. This seems like one of the previous animations which explains three-way handshake of TCP/IP, but the main difference is that, this animation consist of more than one step. Figure 4.28 shows the third step of the animation, and there is still a play button, which tells that the animation is not ended yet.

This animation is designed step-by-step. Each time play button is pressed, another part of the movie plays. There are four steps, so the user should press the play button three times. On the last frame, "back" button appears on the place of play button. Pressing back button results in the animation to begin again. Figure 4.29 shows the end of the animation

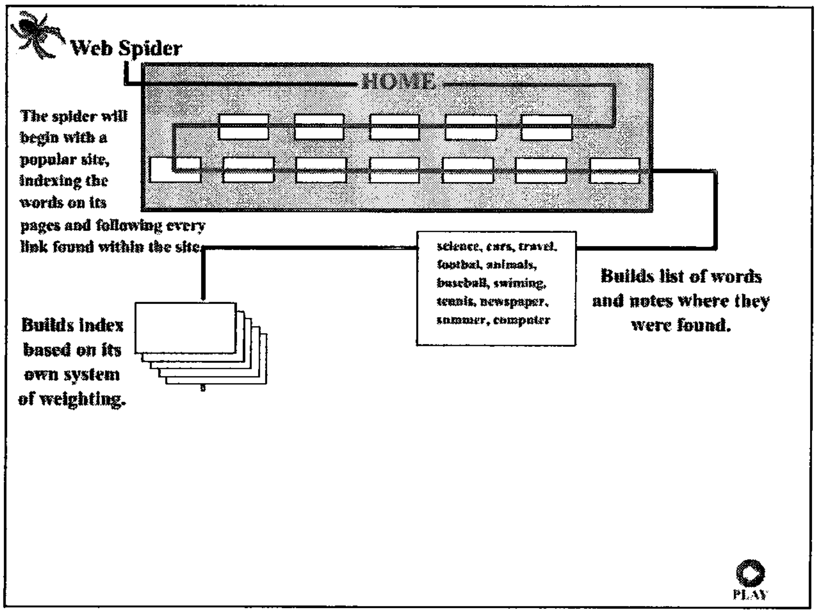

Figure 4.28. Building the index - third step of the animation

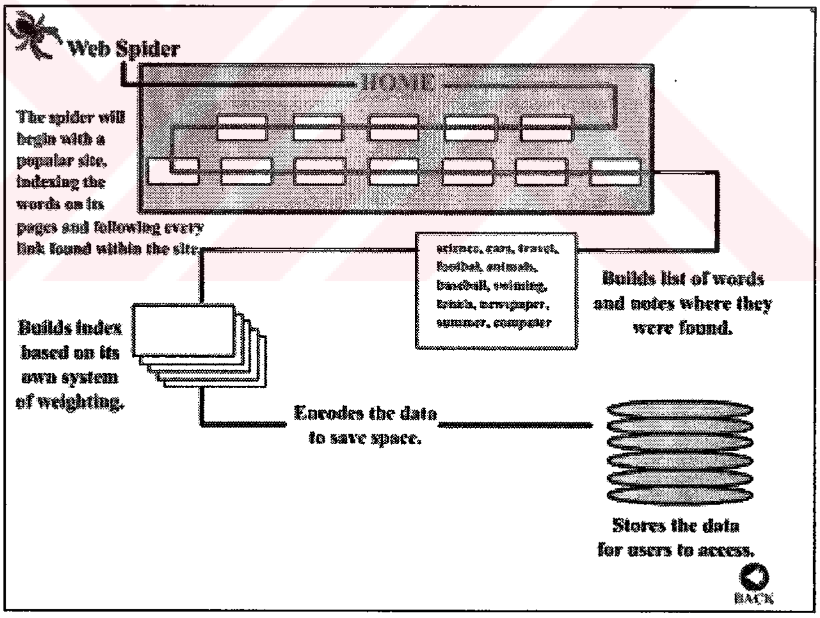

Figure 4.29. Building the index - end of the animation

This animation explains how to build a search engine index step-by-step, and consists of twenty-one layers. Figure 4.30 shows editor view of the animation. Shape-tweening function of Macromedia Flash is frequently used in this animation, especially to draw the moving lines. The beginning and end of a line is constructed, and flash draw the middle parts.

Shape tweening is similar to morphing. With shape tweening, one shape can be made appear to change into another shape over time. To tween a shape (SitePoint, 2003):

i. Click a layer name to make it current.
ii. Select an empty keyframe in the layer, and create the image for the first frame of the sequence
iii. Create a second keyframe where the animation is wanted to end.
iv. Create the image for the last frame of the sequence.
v. Choose Window > Panels > Frame.
vi. For Tweening, select Shape.

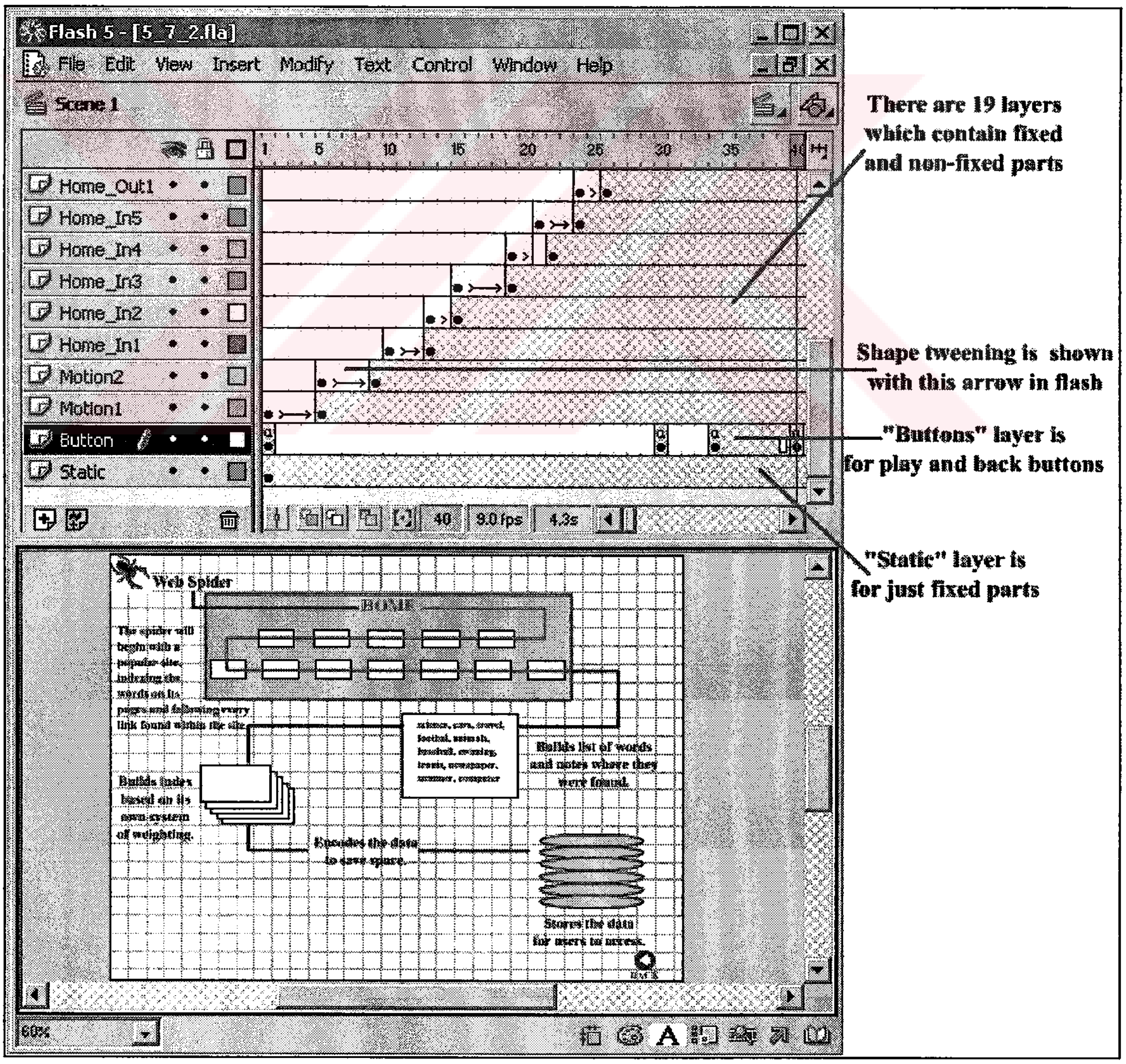

Figure 4.30. Building the index– editor view

In the first step, a web spider, actually a software, begins with a popular site, indexing the words on its pages and follows every link on this page. This is a non-fixed part of the movie, but after this part ends, then it is a fixed part for the remaining of the movie.

There should be more than two layers, if one layer ends its animation, it stays fixed and the next layer begins to run its part. Figure 4.30 shows that a new layer begins after the end of the below layer. There is also a fixed layer, named "Static".

Transitions between steps of the movie are done using flash buttons. There are "play" buttons at the end of each step, and there is a "back" button on the last frame of the movie.

4.4.8. Source, Destination, and Data Packets

The eighth animation is about general communication terms: source, destination and data packet. This animation is taken from chapter four, Networking Basics. In this animation, these three communication terms are explained. Figure 4.31 shows the animation.

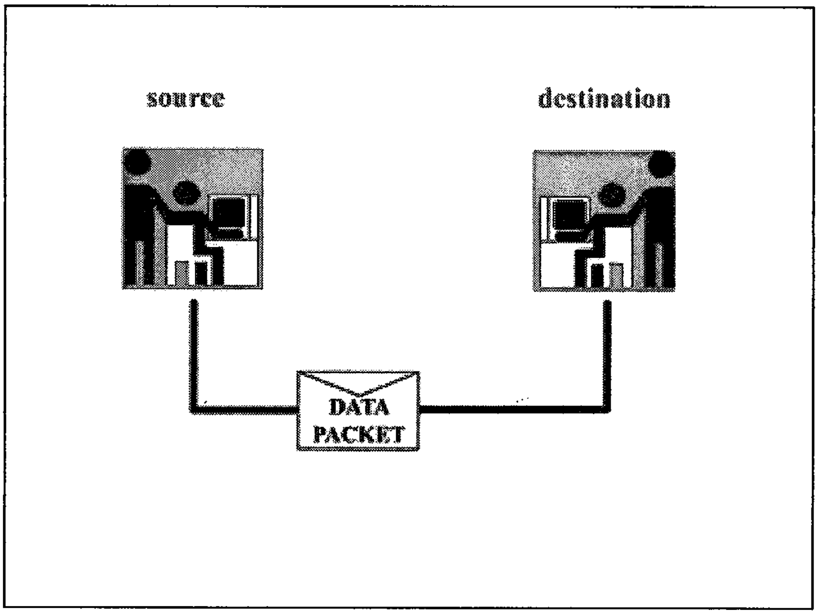

Figure 4.31. Source, destination, and data packets – running animation

In this animation, a data packet is sent from the source to the destination. The black line represents the network cable which connects source and destination. This movie shows the packet while it is traveling on the cable.

The important part of this animation is the movement of the data packet. The packet should travel on a straight line. This can be achieved by frame-by-frame animation, a special keyframe is created for each position of the data packet, and then these frames are played one after the other. This makes the effect as if the packet travels on a straight line.

There is another, more suitable solution for this movement problem. Flash has a feature named "Motion Tweening". Motion tweening is a much quicker way to animate. The concept of tweening is that the designer decides the first and last frames and Flash fills in the middle. Only one object can move in a motion tween on one layer (Clements, 2002).

Figure 4.32 shows the editor view of this animation. There are arrows in the timeline of "Packet" layer. These arrows show places of motion tweening.

Motion tween is done three times in this movie. The black line between source and destination has three straight parts, so motion tween should be done for each of these parts separately.

Tweening is a very important feature of Flash, especially for animations. Without tweening, each frame of an animation should be created as a keyframe. With tweening feature, shape or motion tweening, creating only the first and the last frame is enough.

Similar to shape tweening, motion tweening is another way for animation that takes the position attributes of an object in a start keyframe, and the position and attributes of object in an end keyframe, and calculates all the animation that will occur between the two. The basic distinction between the two types of tweening is that with shape tweening, shapes are used to create the tweening effect, whereas with motion tweening, groups, text, or symbols are used to create the effect (Weil, 2002).

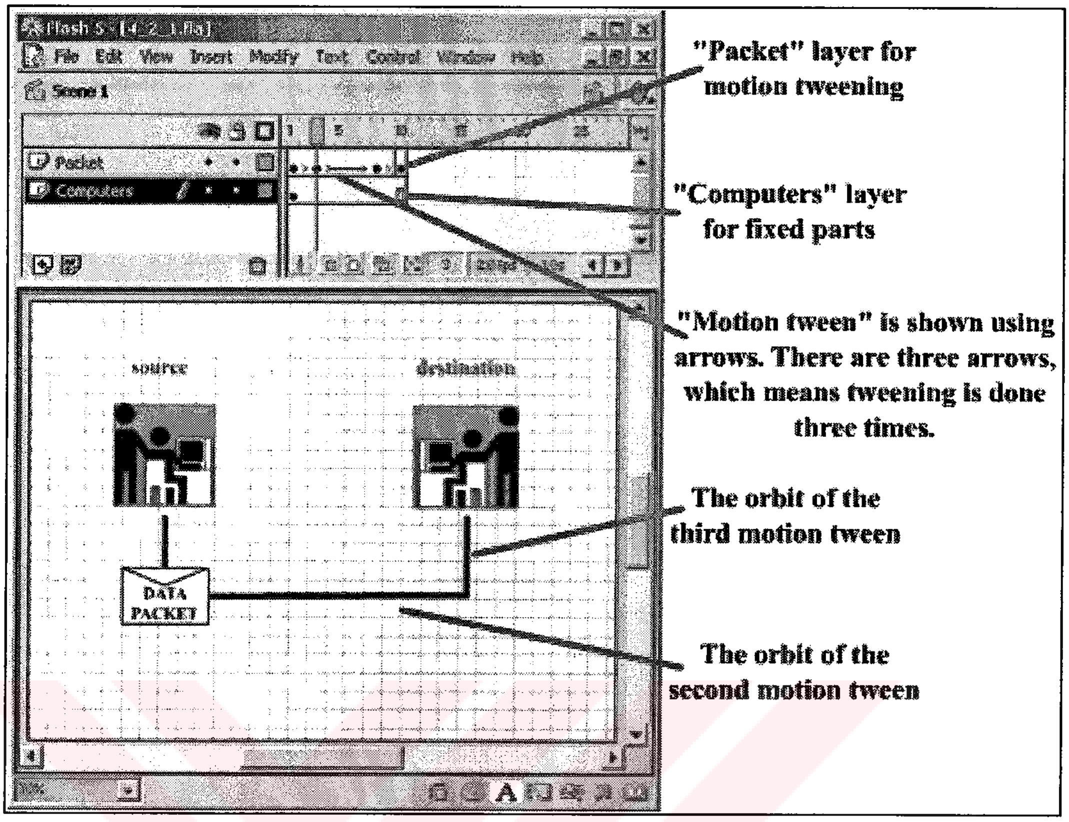

Figure 4.32. Source, destination, and data packets – editor view

To create a motion tween using the Motion Tweening option (Macromedia Livedocs, 2003):

i. Click a layer name to make it the active layer, and select an empty keyframe in the layer where the animation starts.
ii. Create the first frame of the motion tween. Create an instance, group, or text block on the Stage, or drag an instance of a symbol from the Library window. To motion tween an object, it must be converted to a symbol.
iii. Create a second keyframe where the animation ends, then select the ending frame.
iv. Modify the instance, group, or text block in the ending frame. The symbol can be modified by changing the position, scale, rotation, and alpha.
v. Click any frame in the tween's frame span and select Motion from the Tween pop-up menu in the Property inspector (Window > Properties).

4.4.9. Private and Public Key Encryption

The ninth animation is about private and public key encryption. This animation is taken from chapter seven, Network Security. In this animation, these two encryption technologies are explained. Figure 4.33 shows the animation.

This animation is like two separate animations are merged. Button number one initiates private key encryption animation, and button number two initiates public key encryption animation.

To play the private key animation, first, button "1", then "play" button should be selected. Private key animation is designed to explain the algorithm in a step-by-step manner. There are four steps, so "play" button should be pressed three times to reach to the end of the animation.

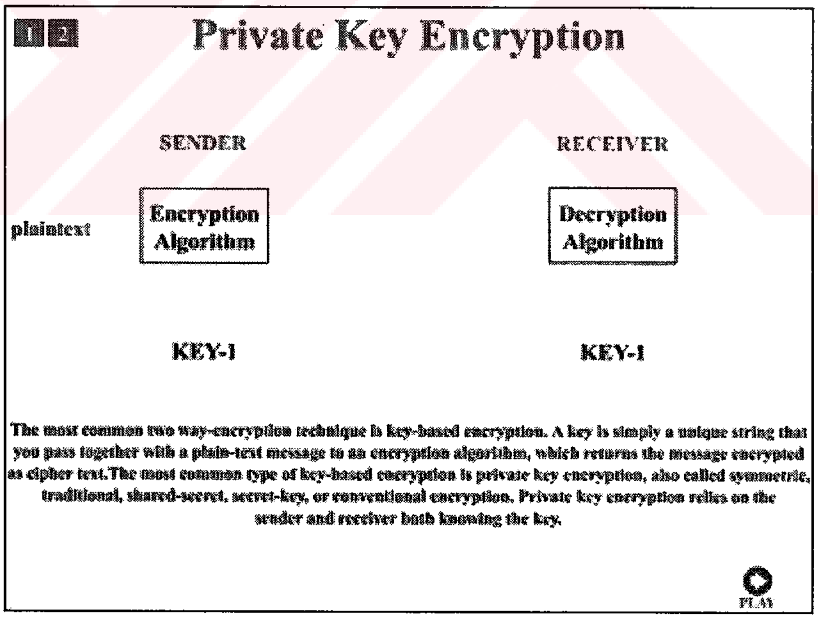

Figure 4.33. Private and public key encryption - start of private key encryption

Figure 4.34 shows the end of the private key animation. There is a "back" button at the last frame which can be used to initiate the animation from the first frame.

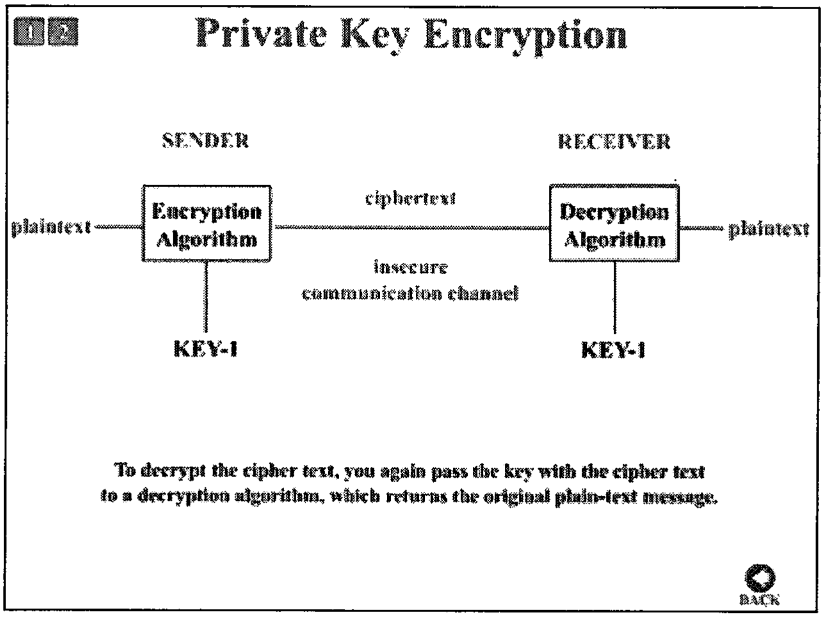

Figure 4.34. Private and public key encryption - end of private key encryption

In order to play the public key animation, first button "2", then "play" button should be selected. Public key animation is designed to explain the algorithm step-by-step. There are four steps, so "play" button should be pressed three times to arrive to the end of the animation.

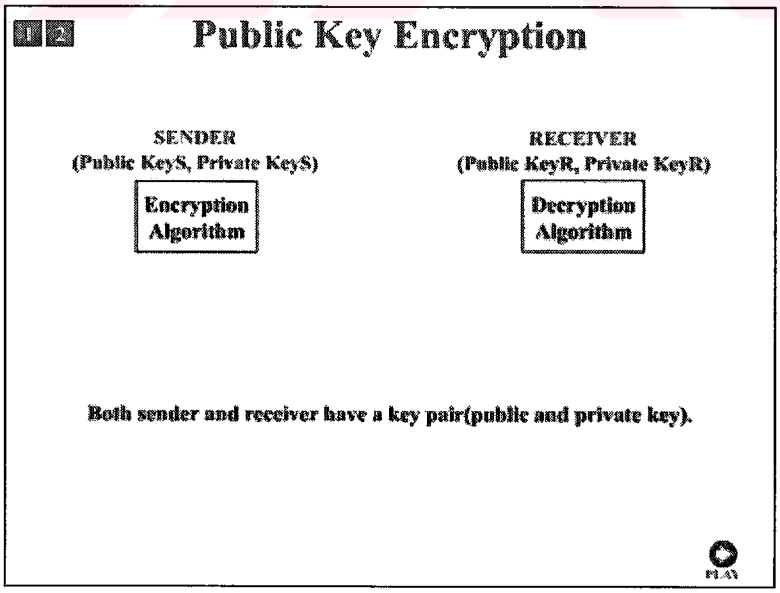

Figure 4.35. Private and public key encryption - start of public key encryption

Figure 4.35 shows the start of the public key encryption. This view is acquired by pressing button “2”. Figure 4.36 shows the end of the public key animation. There is a back button at the last frame which can be used to initiate the animation from the first frame.

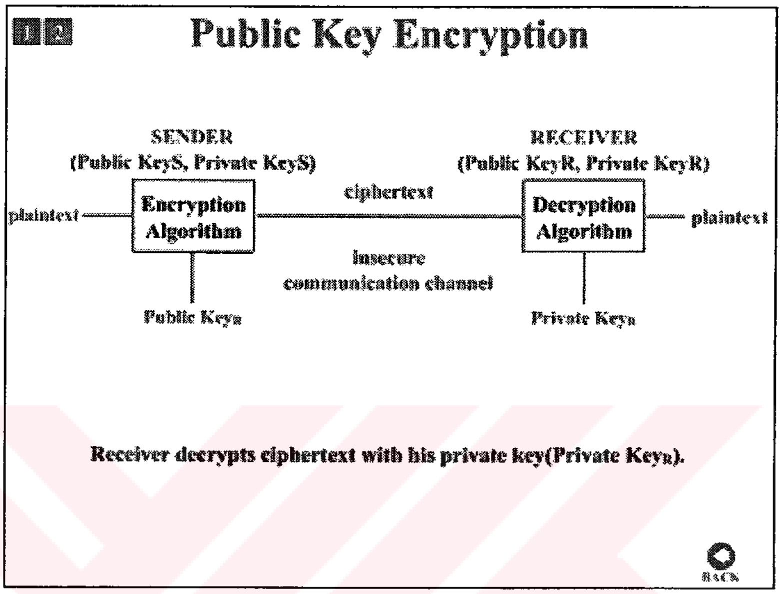

Figure 4.36. Private and public key encryption - end of public key encryption

In this animation, animation techniques in section 4.3.2 and in section 4.3.7 are used in conjunction. Figure 4.37 shows the flash editor view of this animation. There are six layers, two for public key encryption, and two for private key encryption; one for play and back buttons, and one for number buttons.

To make transition between the two animations, number buttons are used. Private key animation is from frame number one to frame number twelve. Public key animation is from frame number thirteen to frame number twenty-seven. ActionScript code written for button”1” invokes frame one, and code written for button“2” invokes frame thirteen.

Both animations are designed in a step-by-step fashion. To control transitions between the steps of each animation, play and back buttons are used. After pressing first “play” button on frame one, the movie plays until frame four and stops. After three steps, the private key animation ends and “back” button appears. Pressing “back” button yields the movie to go to the frame number one. Same rules are valid for public key encryption animation.

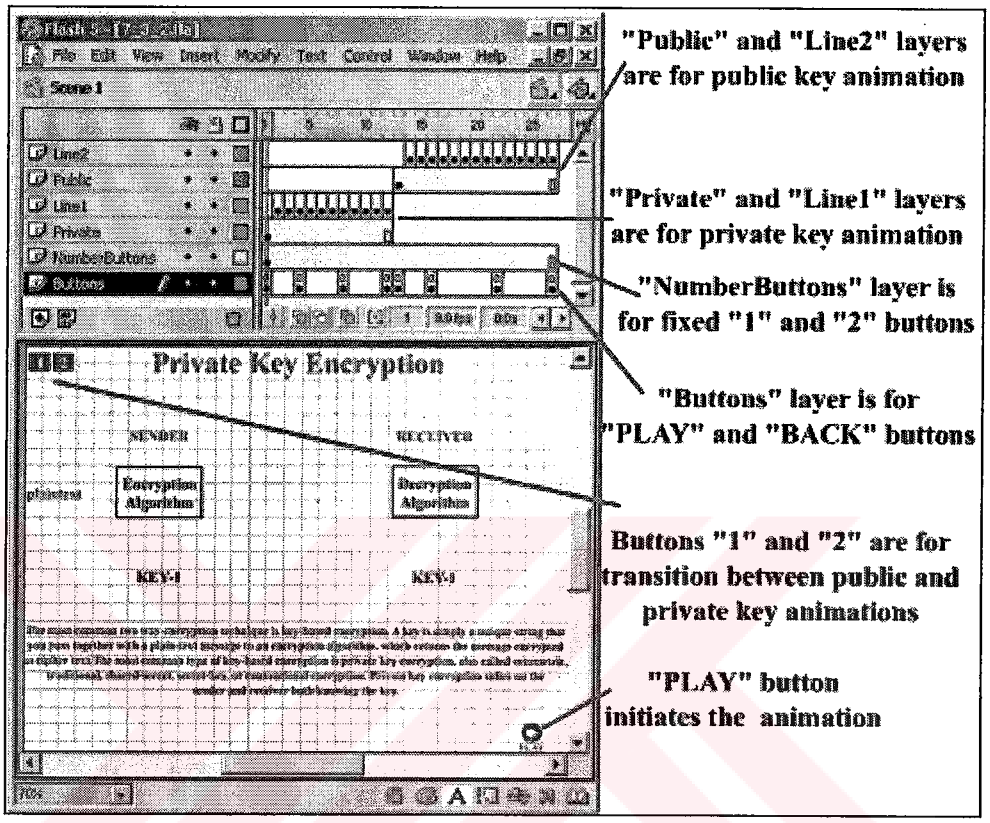

Figure 4.37. Private and public key encryption – editor view

4.4.10. Firewall

The last animation is about network firewalls. This animation is taken from chapter seven, Network Security. In this animation, default behavior of a network firewall is explained. Figure 4.38 shows the start of the animation.

This animation is also prepared on a step-by-step fashion. First step explains what happens to the packet when it originates from the inside network, and step two explains what happens to the packet when it originates from the outside network. Transition between steps is done by pressing “play” button

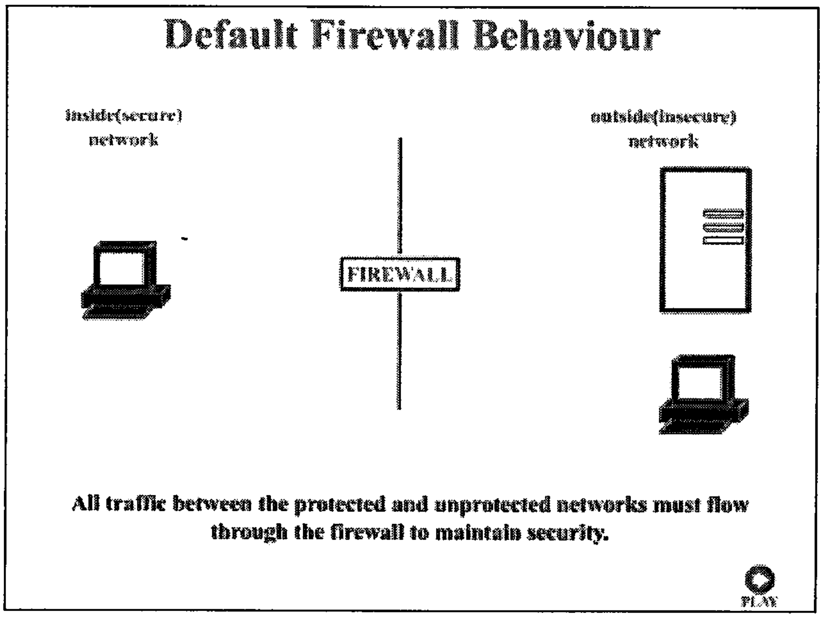

Figure 4.38. Firewall – start of the animation

Frame 4.39 shows the end of the animation. The answer to the packet which originates from inside is allowed through the firewall, but the packet which originates from the outside is not allowed through the firewall.

At the end, a "back" button appears. Pressing this button yields the animation to begin from the first frame.

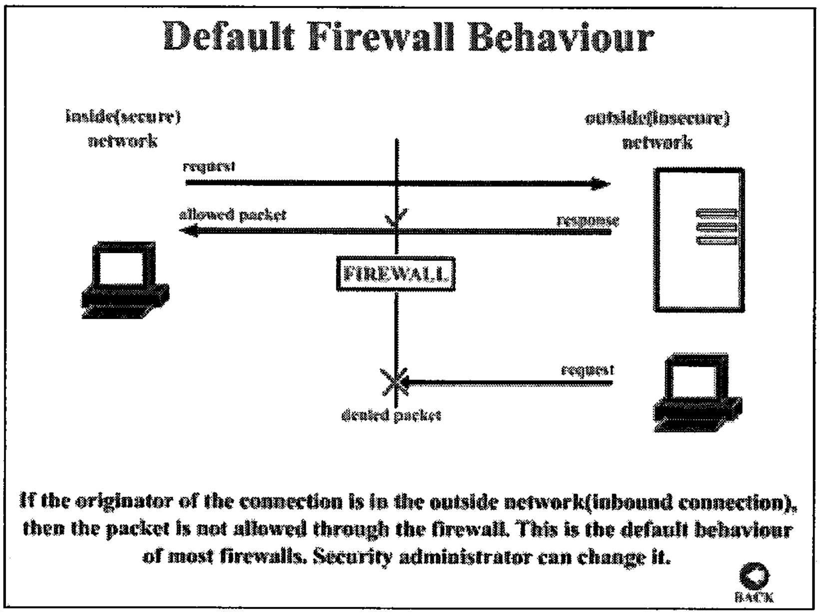

Figure 4.39. Firewall – end of the animation

Figure 4.40 shows the flash editor view of this animation. There are three layers, one for "play" and "back" buttons, one for non-fixed parts like arrows, and one for fixed parts.

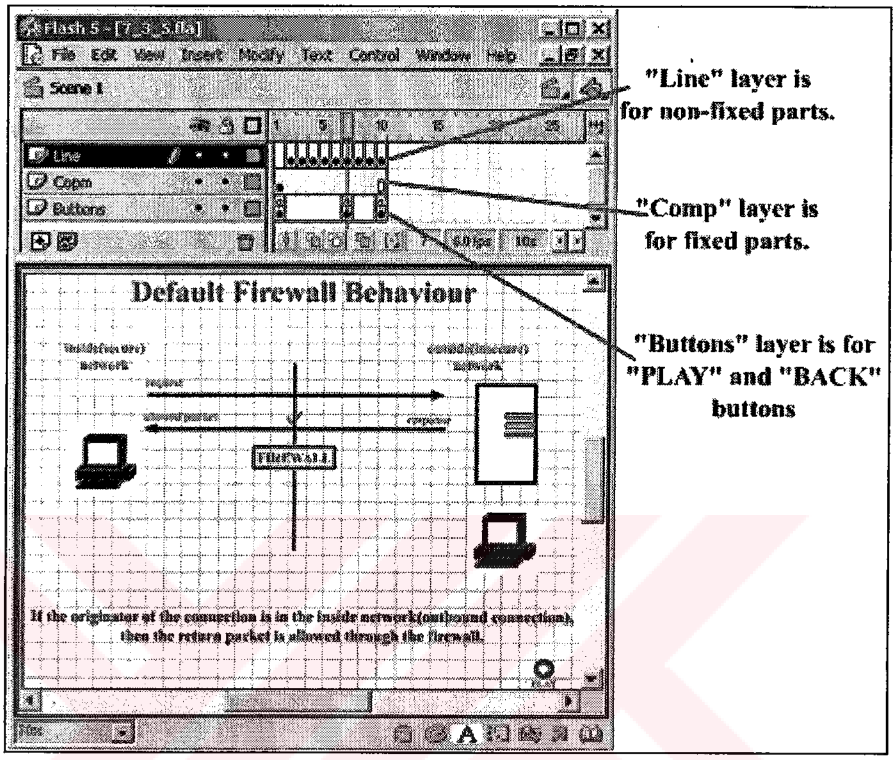

Figure 4.40. Firewall – editor view

Buttons named "play" and "back" are given special tasks using ActionScript. First "play" button plays the movie until frame number seven. At this frame, there is a stop() action and movie stops. "Play" button appears again and wait the user to press on it. When the user press the button, the plays until frame number ten where another stop() action is defined. This is the last frame of the animation and a "back" button appears. If the user wants to view the animation again, "back" button can be pressed. This yields the animation to go back to the first frame.

5. COURSE CONTENT

This section describes the content of the seven online chapters created for this project. Each subsection corresponds to a single online chapter and includes a summary of the lessons and examples of interactive content contained in the chapter. This section does not give any technical information about these interactive content, this was already given in section four.

After giving information about these seven chapters, there is another section that clarifies how to make this electronic content SCORM compatible.

5.1. Chapter One: IT Basics

This chapter discusses the basics of Information Technology. There are four sections and twenty-two subsections in this chapter. These four sections are: Getting Started in IT, Basic Mathematics for IT, Windows Desktop Environment, Overview of Software Applications.

The chapter covers different computer types and software applications. A brief overview of the Internet and the methods used in number conversions, including binary to decimal and decimal to binary are also included.

5.1.1. Getting Started in IT

The first section of Chapter One has four subsections:

- Computer Systems and Programs
- Computer Types
- Connecting Computer Systems
- Birth of the Internet

These subsections explain computer systems and programs, hardware and software components of a computer system, computer types like pc and mainframe, advantages of networked computers over standalone computers and a short history about Internet.

5.1.2. Basic Mathematics for IT

The second section of Chapter One has six subsections:

- Terminology
- Boolean Logic Gates
- Decimal and Binary Systems
- Decimal to Binary Conversion
- The Hexadecimal Number System
- Conversion

These subsections explain basic mathematics' concepts for information technologies. Definitions like bit, byte, kilobyte, bit-per-second; logic gates like "and", "or", "xor"; decimal number system, binary number system, hexadecimal number system, and conversions between these number systems are included in this section. Especially, binary number system is very important in computer engineering.

5.1.3. Windows Desktop Environment

The third section of Chapter One has seven subsections:

- Starting, Shutting down, and Restarting Microsoft Windows
- Windows Explorer
- The Desktop
- Working with Icons
- Recognizing an Application Window
- Switching between Windows
- Viewing the Basic System Information of a Computer

Windows operating system is one of the most common operating systems, especially for those who begin using computer very recently. For this reason, this section gives basic information about windows OS. This section covers starting, shutting down and restarting Microsoft Windows; running and using Windows Explorer; components of desktop like start menu, system tray, and quick launch; working with icons like creating, moving and renaming the icons; and viewing basic system information.

5.1.4. Overview of Software Applications

The last section of Chapter One has five subsections:

- Word Processors
- Spreadsheets
- Databases
- Presentation Applications
- Web Browser and E-mail

This section is about one of the most used application group: office programs. It gives an introduction to word processor, spreadsheet, database and presentation applications. At the end, some information about email and web browsers is also given, but not in detail. More features about e-mail and web browsers will be given in Chapter five, Internet Basics.

5.2. Chapter Two: Computer Basics

This chapter discusses the basics of Computers. There are seven sections and twenty-eight subsections in this chapter. These seven sections are: Boot Process, Hardware Components, Display Components, Connector Components, Storage Components, Network Components and System Resources.

This chapter is about how computers work. Firstly, an idealized computer and computer boot process is explained. The boot process includes initializing and testing the system, loading the OS, and the boot sequence that is required to operate the computer.

Then, Computer Hardware like CPU, motherboard, memory, BIOS, etc. are explained in detail.

5.2.1. Boot Process

The first section of Chapter Two has four subsections:

- An Idealized Computer
- Initializing the System Hardware
- Loading the Operating System
- The Boot Sequence

This section firstly explains an idealized computer which has a CPU, memory, and interfaces which are connected through a common bus. In the second subsection, information about the responsibilities of BIOS and different BIOS types is given. In the third subsection, information about the OS loading process is given. Screenshots from different OS loading processes are given as a flash animation. In the last subsection, boot sequences like cold boot and warm boot are given.

5.2.2. Hardware Components

The second section of Chapter Two has nine subsections:

- Computer Case
- Power Supply
- Cooling Systems
- The Motherboard
- The CPU
- BIOS
- Expansion Slots
- Bus Types
- Memory

In the first subsection, computer case types like tower and desktop explained and case images are given using flash. Secondly, power supply types are explained and different power supply images are shown using a flash animation. In the third subsection, information about cooling system is given and the important parts of a cooling system are explained with a flash animation. In the fourth subsection, motherboard components are introduced. Fifth one gives information about the CPU; and different CPU types are displayed using flash. Sixth subsection is about BIOS types and responsibilities. Then, expansion slots like ISA, AGP and PCI introduced. Bus types are explained after that. Lastly, memory types and memory duties are clarified.

5.2.3. Display Components

The third section of Chapter Two has two subsections:

- Monitors
- Video Cards

First subsection explains basic terms about monitors like pixel, dot pitch, refresh rate, color depth, Video RAM, resolution, screen size and display colors. Terms like VGA, SVGA and XGA are also explained.

Last part is about video cards. The specifications of a video card are given, and parts of a video card are explained using a flash animation.

The flash animation about video cards is shown in Figure 5.1. This animation is prepared using techniques introduced in section four. The following animation includes three different video card examples. Only the first video card and its components are shown in Figure 5.1.

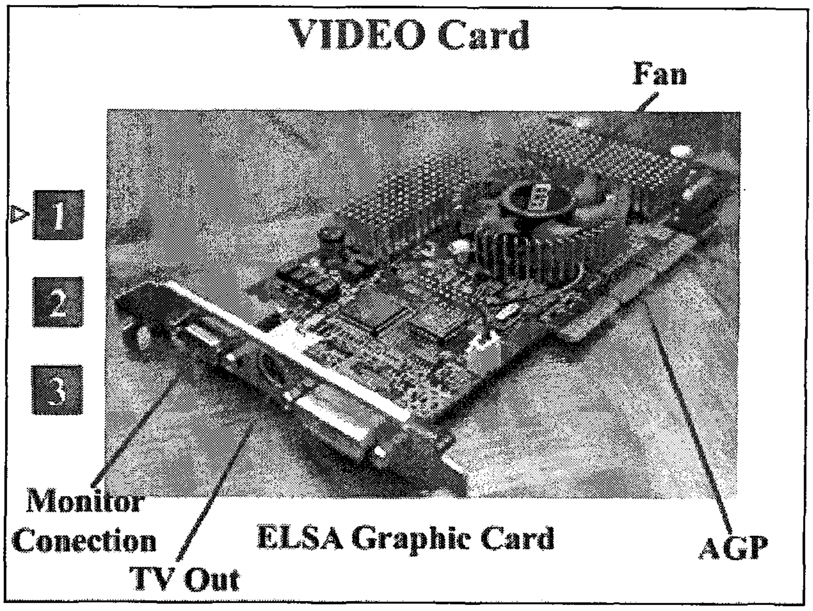

Figure 5.1. Video cards

5.2.4. Connector Components

The fourth section of Chapter Two has four subsections:

- Serial and Parallel Ports
- PS/2 ports
- Universal Serial Bus (USB) and FireWire
- IDE, EIDE, Ultra, and SCSI controllers

First section is about serial and parallel I/O ports. Differences between serial and parallel ports are given. Secondly, PS/2 mouse and keyboard ports are explained. Third one is about USB and FireWire ports. Data transfer rates about USB, USB 2.0 and FireWire are given. Especially, data rate of USB 2.0, with 450 Mbps, and FireWire, with 400 Mbps are very high rates. Figure 5.2 shows a flash animation about these ports. Only the third image is displayed in figure.

Last part is about IDE, EIDE, Ultra, and SCSI controllers. These are hardware interfaces used to connect hard disks, CD-ROMs, and tape drives to a PC. There is a flash animation about there controllers. SCSI chain is also explained.

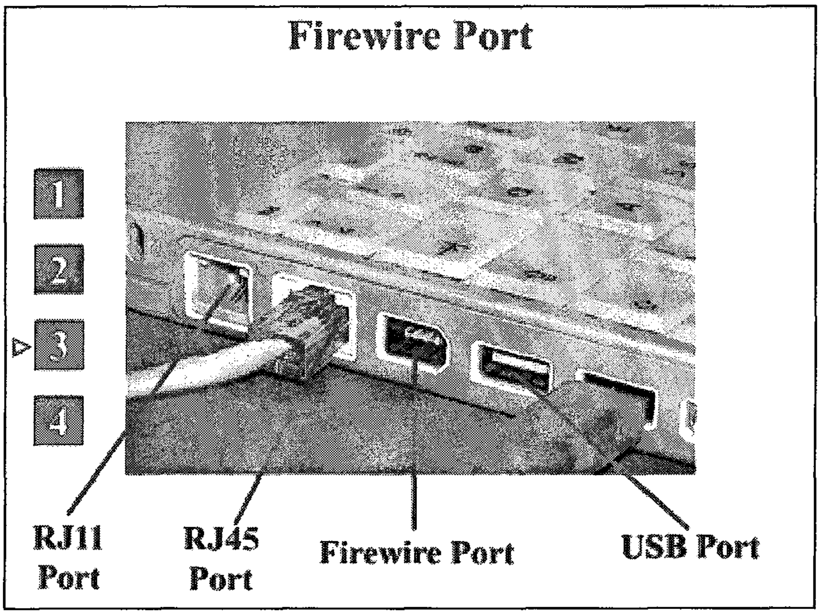

Figure 5.2. USB and firewire

5.2.5. Storage Components

The fifth section of Chapter Two has three subsections:

- Floppy drives
- Hard Drives
- CD-ROMs

Firstly, floppy disk drives (FDD) are explained. Currently, 3.5" floppy diskettes are used. They only hold 1.44 MB of information. Second part explains hard drives, main storage medium of the computer. Parts of a HDD are made clear. At the end of this section, information about CD-ROM and CD-ROM speed ratings are given.

5.2.6. Network Components

The sixth section of Chapter Two has two subsections:

- Modems
- Network Interface Card (NIC)

First part is about modems. A modem is a device that converts the digital data used by computers into analog signals, suitable for transmission over a telephone line, and converts the analog signals back to a digital signal at the destination. It uses a dialup networking connection.

Last part is about NIC. NIC is used to connect a local computer to a group of other computers. NICs connect computers so they can share data and resources in a networked environment. There is also a flash animation which displays modem, NIC and UTP patch cord with RJ45 connector.

5.2.7. System Resources

The seventh section of Chapter Two has four subsections:

- Overview
- Interrupt requests (IRQs)
- DMA
- Input/Output (I/O) Addresses

First of all, an overview about system resources is given. System resources refers to the mechanisms used to interface, communicate, and control individual device adapters along with the serial, parallel, and mouse ports. System resources are shared between the different hardware components of the computer system that need to communicate with the CPU.

Then, IRQ is explained. Various hardware devices may want to tell the CPU that there is information available that is ready for transfer. The devices indicate this by making an interrupt request, or IRQ.

Third one is about DMA. Direct Memory Access channels allow devices to bypass the processor and directly access the computer memory. Last one is about I/O addresses. An I/O port number is a memory address where data is temporarily stored as it moves in and out of the devices.

5.3. Chapter Three: Operating System Basics

This chapter discusses the basics of the operating systems. There are six sections and twenty-six subsections in this chapter. These six sections are: The Operating System, Overview of PC Operating Systems, Disk Operating System (DOS), Microsoft Windows, UNIX and Linux, and Memory Management.

The operating system is the program that manages computer operations. This chapter describes the components, functions, and important terminology related to an operating system. Basics of the most common operating systems, like DOS, Windows, and Unix/Linux are covered in this chapter.

5.3.1. The Operating System

The first section of Chapter Three has three subsections:

- Components of an Operating System
- Operating System Functions
- Operating System Types and Basic Terminology

Firstly, an introduction to operating system is given. An operating system is a software program that provides an interface between the user and the computer and manages thousands of applications. This part also lists three main components of an operating system: user interface, kernel and file management. Details about these three components are given in the second section of the chapter.

In the second part, operating system functions are introduced. These functions are file and folder management, application management and computer hardware control.

Last part of this section introduces basic terminology like multi-user, multitasking, multiprocessing and multithreading. This part also lists the most popular operating systems. Microsoft OS, Macintosh OS and UNIX are listed.

5.3.2. Overview of PC Operating Systems

The second section of Chapter Three has three subsections:

- The Kernel
- The User Interface
- The File System

These are the main components of an operating system. The kernel is the innermost portion of the operating system that handles hardware management. The UI is the component of the OS that the user interacts with. It acts like a bridge between the user and the kernel. The file system of an OS determines the way files are named and how and where they are placed on storage devices, such as hard disks.

5.3.3. Disk Operating System (DOS)

The third section of Chapter Three has six subsections:

- Overview
- DOS File Structure
- Overview of Basic DOS Commands
- Creating a DOS Boot Disk
- Booting the System with a DOS Disk
- DOS Configuration Files

First subsection is an overview of DOS. Microsoft developed the Disk Operating System (DOS) in 1981. DOS, which is sometimes called MS-DOS, was designed for the IBM PC. Secondly, DOS file structure is introduced. Third part is about most common DOS commands.

Fourth part is about DOS boot disk. Three important files, named command.com, io.sys, and msdos.sys are introduced. Fifth part explains booting with DOS system diskette which is prepared in fourth part. Master boot record (MBR) is also introduced. At the end of this section, DOS configuration files are brought in. These files are config.sys and autoexec.bat. These files can be included in DOS boot up process.

5.3.4. Microsoft Windows

The fourth section of Chapter Three has five subsections:

- Windows 3.1
- Windows 9x
- Windows NT and Windows 2000
- Windows XP
- Windows CLI

This section covers all versions of Microsoft Windows operating systems beginning from Windows 1.0 released in 1985. Specifications of each version are given. There are flash animations for each subsection which display desktop arrangements, login screens and logos of each version. Figure 5.3 shows one of these flash animations related to Windows 1.0.

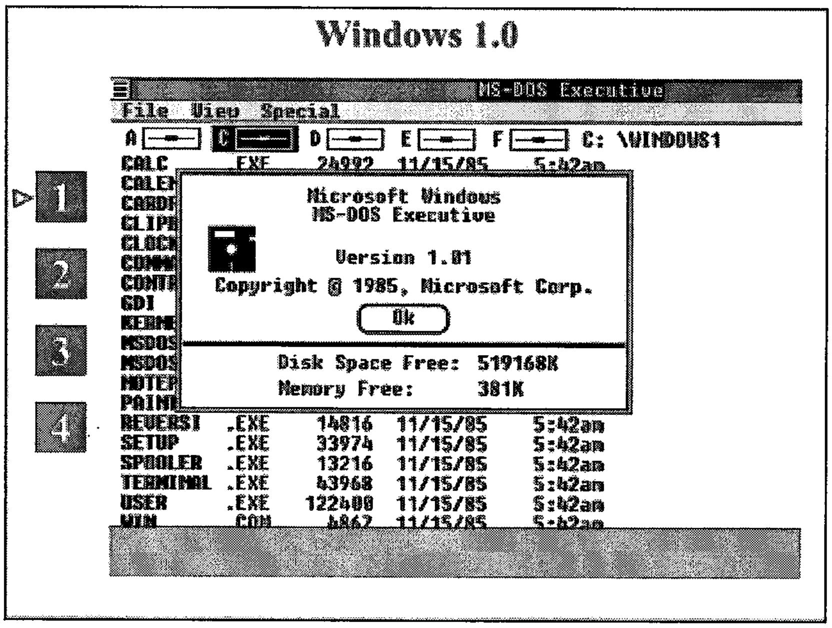

Figure 5.3. Windows 1.0

5.3.5. UNIX and Linux

The fifth section of Chapter Three has five subsections:

- Origins of UNIX
- Origins of Linux
- Linux/UNIX GUI
- Linux/UNIX CLI
- Linux and UNIX System Configuration Tools

This section explains the multi-user and multitasking operating systems, UNIX and Linux. It gives information of networking in these operating systems; graphical user interfaces like GNOME and KDE, text based interfaces like Bourne shell, Bash shell, and system configuration tools like linuxconf for Linux and UNIX, and Admintool for SUN Solaris operating system.

5.3.6. Memory Management

The last section of Chapter Three has four subsections:

- Memory Types
- Virtual Memory
- Memory Conflicts
- Real versus Protected Mode Memory

Last section is about memory management. In the first part, concepts like conventional memory, upper memory, expanded memory, extended memory and high memory are explained. Secondly, virtual memory concept and virtual memory configuration are made clear. Third part is about memory conflicts. Causes of memory conflicts are explained and memory conflict messages in Windows operating system are given. At the last, real mode memory versus protected mode memory are compared.

5.4. Chapter Four: Networking Basics

This chapter discusses the basics of the networking. There are four sections and eighteen subsections in this chapter. These four sections are: Network Types, General Communication Terms, OSI Reference Model, and TCP/IP Model.

This chapter starts with an overview of networking. Definition of networking, and network types (LAN, WAN, peer-to-peer, and client/server) are given. Then, OSI and TCP/IP Models are discussed.

5.4.1. Network Types

The first section of Chapter Four has three subsections:

- Overview
- Peer-To-Peer and Client-Server
- LAN/WAN

In the first part, an overview of network is given. By definition, a computer network has two or more devices linked together for the purpose of sharing information and resources. Secondly, peer-to-peer and client-server network types are explained. A flash animation is prepared to clarify these two concepts. At the end, definition of LAN and WAN are given.

5.4.2. General Communication Terms

The second section of Chapter Four has three subsections:

- Source, Destination, and Data Packet
- Media
- Protocols

This section gives brief information about source, destination and data packet concepts. Then media is defined and with a flash animation media types like coaxial cable, UTP cable and fiber optic cable are introduced. At the last part, protocol is described. Communication Protocol is a standard designed to specify how computers interact and exchange messages.

5.4.3. OSI Reference Model

The third section of Chapter Four has seven subsections:

- Organizations dealing with Standards
- The evolution of ISO Networking Standards
- The purpose of the OSI reference model
- Seven Layers of OSI Reference Model
- Characteristics of the OSI Layers
- The functions of each Layer
- Information Exchange

This section is about OSI reference model. To address the problem of networks being incompatible and unable to communicate with each other, the International Organization for Standardization (ISO) researched network schemes and constitute a network model that would help vendors create networks that would be compatible with, and operate with, other networks. The Open System Interconnection (OSI) reference model describes how information from a software application in one computer moves through a network medium to a software application in another computer.

There are seven layers in the OSI reference model. This section explains advantages and characteristics of the layered model. It also explains the function of each of these layers in detail.

At the end of this section, information exchange between source and destination systems is also clarified. This exchange is explained using a flash animation which shows what happens to a packet at each layer.

5.4.4. TCP/IP Model

The fourth section of Chapter Four has six subsections:

- History of TCP/IP and Internet
- Protocol Architecture
- TCP/IP Layers
- Protocol Graph
- Transmission Control Protocol
- Comparing OSI with TCP/IP

This section explains TCP/IP model, and compares TCP/IP with OSI reference model. Although the OSI reference model is universally recognized, the historical and technical open standard of the Internet is Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP is a set of protocols developed to allow cooperating computers to share resources across a network.

The four layer of TCP/IP model are clarified. These layers are explained using a flash animation. At the end of the section, the connection establishment method of TCP/IP, named "Three-way Handshake" is described.

5.5. Chapter Five: Internet Basics

This chapter discusses the basics of the Internet. There are ten sections and twenty-seven subsections in this chapter. These ten sections are: Internet, IP Addressing, WWW, E-MAIL, FTP, TELNET, Search Engine, Newsgroups, Chat and "Virus, Trojan, Worm"

In this chapter, some basic protocols and applications on Internet are explained. This chapter is created to teach just the basics of internet, such as electronic mail, World Wide Web, file transfer protocol, telnet and newsgroups.

5.5.1. Internet

The first section has just one subsection. It gives an overview about Internet. The Internet is a network of networks, linking computers to computers through the TCP/IP protocol. Each runs software to provide or serve information and/or to access and view information. The Internet is the transport vehicle for the information stored in files or documents on another computer.

5.5.2. IP Addressing

The second section of Chapter Five has four subsections:

- Overview
- IP Version 4 Limitations
- IP Address Classes
- DNS (Domain Name System)

At first step, a definition of IP addressing is given. When using the Internet, every computer has a unique number which identifies this computer to other computers on the Internet. This unique number is known as the computer's IP (Internet Protocol) Address.

Secondly, IP version four and IP version six are compared and some web links about IP version six are given.

Third subsection is about IP address classes. All address classes are explained. There is also a flash animation which compares A, B and C address classes.

At the last step, DNS is clarified. The requirement for DNS system is explained. This subsection also has a flash animation which makes clear giving IP address parameters to a computer. It shows step-by-step an IP configuration.

5.5.3. WWW

The third section of Chapter Five has three subsections:

- Overview
- Web Browsers
- A Web Transaction

At first, a brief introduction to WWW is given. World Wide Web is the fastest growing application used in Internet. WWW is a multimedia system using the client/server technology. A client application allows a normal user to access the information contained in another computer connected to Internet.

Second subsection explains web browses. Web browser is the software program used to access the World Wide Web, the graphical portion of the Internet.

Third part explains a web transaction step-by-step. This step-by-step explanation is supported with a flash animation shown in Figure 5.4.

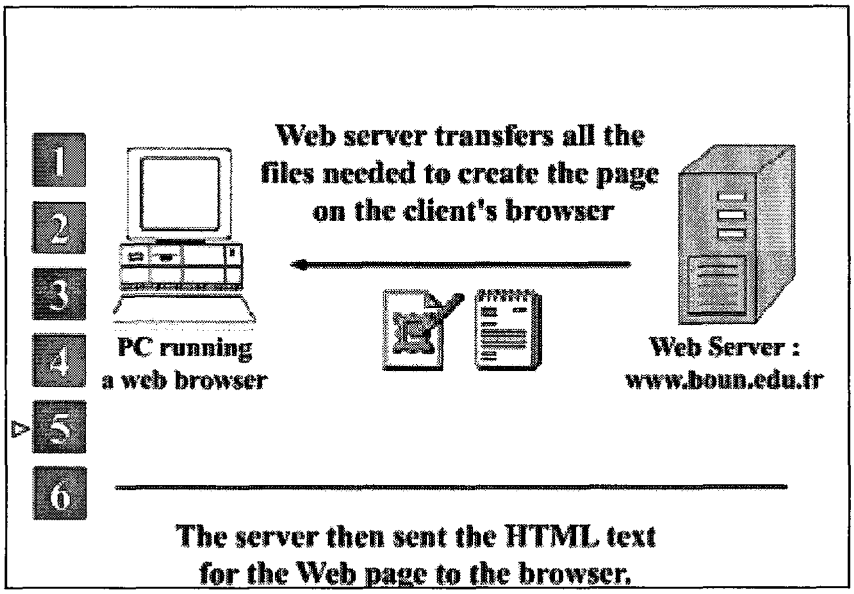

Figure 5.4. Fifth step of a web transaction

5.5.4. E-MAIL

The fourth section of Chapter Five has four subsections:

- Overview
- Address Structure
- Anatomy of an E-Mail Message
- Sending an E-mail

First, an overview about e-mail is given. E-mail is an electronic message sent from one computer to another. Second part explains each part of an e-mail address. This part is supported with a flash animation. Third subsection clarifies parts of an e-mail message. At the end of this section, some concepts like composing an e-mail, setting message importance, sending messages, opening messages, replying and forwarding are introduced.

5.5.5. FTP

The fifth section of Chapter Five has three subsections:

- Overview
- Basic FTP Commands
- GUI Clients

First subsection gives a definition of FTP protocol. FTP stands for File Transfer Protocol, which is simply a method of transferring files on the Internet. Using FTP, sending (upload) and receiving (download) files to and from FTP sites on the Internet are possible.

Second part introduces basic FTP commands, like put, mput, get, mget, binary, ls, open, quit and bye. A sample FTP transaction is shown using a flash animation.

Last subsection is about FTP GUI clients. Three of them, named "Cute FTP", "WS-FTP" and "FTP Now" are introduced. Their interfaces are also shown using a flash animation.

5.5.6. TELNET

The sixth section of Chapter Five has three subsections:

- Overview
- Using Telnet
- Telnet vs. SSH

First part is an overview of TELNET protocol. TELNET is the main Internet protocol for creating a connection with a remote machine. Second part explains how to telnet a remote machine, and last part compares TELNET protocol and SSH protocol. SSH is a protocol that allows users to access remote computers with a reasonable amount of security.

5.5.7. Search Engine

The seventh section of Chapter Five has three subsections:

- Overview
- Building the Index
- Building a Search

The Internet can be the quickest and least expensive way to find information, as long as search engines are used efficiently. This section explains how to use search engines powerfully.

At the first step, an overview of the search engines is given. Internet search engines are special sites on the Web that are designed to help people find information stored on other sites.

Second part explains how a search engine builds its index. This subsection is supported with a flash animation. At the end, building search queries are explained. Most often used Boolean operators are introduced, and with a flash animation, some example queries are shown.

5.5.8. Newsgroups

The eighth section of Chapter Five has two subsections:

- Overview
- The Newsgroup Process

Firstly, an overview about newsgroups is given. A newsgroup is a continuous public discussion about a particular topic. Second part explains newsgroup process. A newsgroup begins on a single news server, but is eventually replicated to hundreds or thousands of other servers. News servers provide the infrastructure that makes newsgroups work. This part is supported with a flash animation showing how to set up a news server account. Figure 5.5 shows the fourth step of this flash animation.

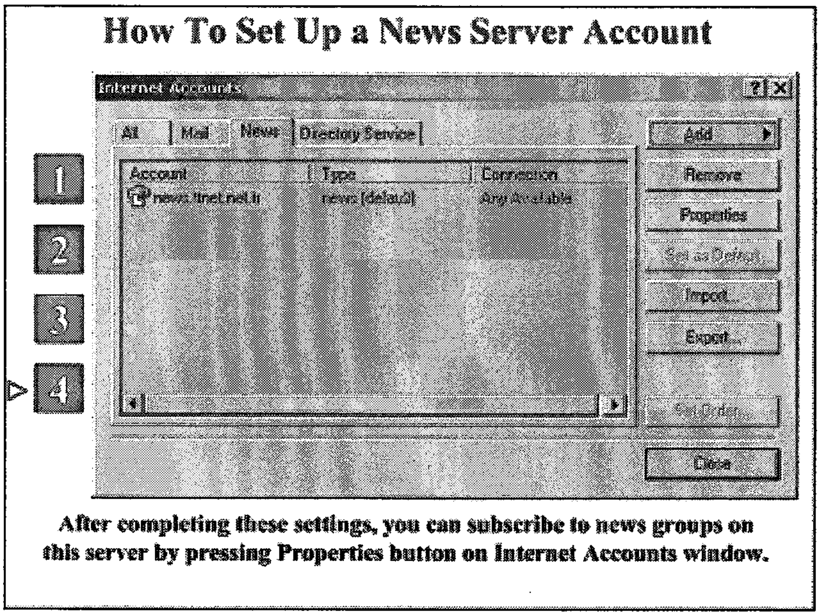

Figure 5.5. Fourth step of news server account setting

5.5.9. Chat

The ninth section has just one subsection. It gives an overview about chat. Internet Relay Chat is one of the many popular methods of communicating over the Internet. Chat is a text-based conferencing system.

5.5.10. Virus, Trojan, Worm

The last section of Chapter Five has three subsections:

- Virus
- Trojan
- Worm

Firstly, virus is explained. The word virus is often used very generally when normal computer users talk about unwished programs they have got into their computers.

Second subsection is about Trojan. A Trojan is a program, hidden in something looking friendly or useful, but makes it possible for a hacker to take full control over the computer.

At the end, worm is clarified. Programs that are using the security holes of the operating system or working services of computer to infect it automatically are called as worm.

5.6. Chapter Six: WWW Basics

This chapter discusses the basics of the WWW. There are five sections and thirty-four subsections in this chapter. These five sections are: HTML, CGI, Java Script, ASP and PHP. In this chapter, introduction to various Web technologies is given.

5.6.1. HTML

The first section of Chapter Six has seven subsections:

- Overview
- Tag Syntax
- Tags
- Frames
- The <frame> Tag and Frame Targets
- Tables
- Colors

This section explains HTML and its components. HTML (Hypertext Markup Language) is the language used to encode World Wide Web documents. It is a document-layout and hyperlink-specification language that defines the syntax and placement of special, embedded directions that aren't displayed by a Web browser, but tell it how to display the contents of the document, including text, images, and other supported media.

5.6.2. CGI

The second section of Chapter Six has five subsections:

- Overview
- Environment Variables
- Sample CGI Programs in PERL
- Form Input
- The GET and POST Methods

The first subsection gives an overview about CGI. The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers. This part is supported with a flash animation which is shown in Figure 5.6. This is a simple diagram of CGI.

Second part explains CGI environment variables like GATEWAY_INTERFACE, SERVER_NAME, REMOTE_HOST, etc. Third part gives some examples about CGI programs written in PERL.

Fourth part clarifies HTML forms. This part is supported with a flash animation which shows a simple HTML code for forms. Last subsection explains GET and POST methods, their advantages and disadvantages.

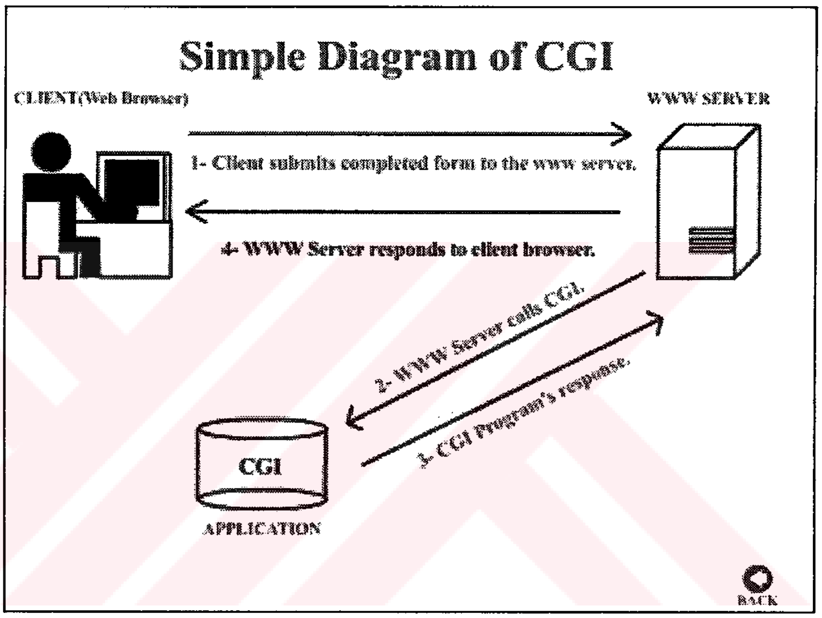

Figure 5.6. Simple diagram of CGI

5.6.3. Java Script

The third section of Chapter Six has seven subsections:

- Overview
- Variables
- Numbers and Strings
- Boolean Values
- Functions
- Objects
- Client-Side JavaScript Examples

First part gives an overview about JavaScript. JavaScript is a scripting language that can be used to extend the capabilities of HTML pages on the World Wide Web. Second part explains variables. Variables are names that have values assigned to them. Third part gives information about numbers and strings.

Fourth subsection is about Boolean values. A Boolean value represents a "truth value", it says whether something is true or not. Fifth part explains functions in JavaScript. A function is a piece of JavaScript code that is defined once in a program and can be executed, or invoked, many times by the program. Then, objects are explained, and at the end of this section, two JavaScript examples are given. These examples are shown using a flash animation.

5.6.4. ASP

The fourth section of Chapter Six has ten subsections:

- Overview
- Built-in Objects
- Application Object
- ASPError Object
- Request Object
- Response Object
- Server Object
- Session Object
- Quick Reference
- Examples

This section is about ASP. Microsoft's Active Server Pages (ASP) technology provides a framework for building dynamic HTML pages which enable Internet and Intranet applications to be interactive.

First part gives an overview. Then, the important ASP objects are explained. At the end, a quick reference is given and soma example codes written in ASP are displayed.

5.6.5. PHP

The fifth section of Chapter Six has five subsections:

- Overview
- Basics
- Functions
- Arrays
- I/O

This section is about PHP. PHP (which stands for "PHP: Hypertext Preprocessor") is an open-source server-side scripting language which can be embedded within HTML. PHP offers a simple and universal solution for easy-to-program dynamic Web pages.

This section explains basics of the PHP with examples. Functions, arrays and I/O model are also included.

5.7. Chapter Seven: Network Security

This chapter discusses the basics of the network security. There are three sections and sixteen subsections in this chapter. These three sections are: Network Security Policy, Common Network Threats and Implementing Security.

Network security is a complicated subject, historically only tackled by well-trained and experienced experts. However, as more and more people become connected to internet, an increasing number of people need to understand the basics of security in a networked world. This chapter was written with the basic computer user and information systems manager in mind, explaining the concepts of network security, understand risks and how to deal with them.

5.7.1. Network Security Policy

The first section of Chapter Six has five subsections:

- Overview
- Need for Network Security Policy
- Username and Password Standards
- Virus Protection Standards
- Server Room Security

A network security policy defines the organization's expectations of proper computer and network use and the procedures to prevent and respond to security incidents. This section describes network security policy.

5.7.2. Common Network Threats

The second section of Chapter Seven has five subsections:

- Risks of Network Connectivity
- Port Scanning
- Denial of Service Attack
- IP Address Spoofing
- IP Address Sniffing

This section explains common network threads. Before a network security policy can be established, a risk analysis has to be studied. Risk analysis is the process of identifying what you need to protect, what you need to protect it from, and how to protect it. It is the process of examining all of your risks, and ranking those risks by level of severity. In this section, common attack types are explained. Each thread is supported with a flash animation which explains the thread in a step-by-step fashion.

5.7.3. Implementing Security

The last section of Chapter Seven has six subsections:

- File Encryption
- Private and Public Key Encryption

- Secure Socket Layer (SSL)
- IP Security (IPSec)
- Firewall
- OS Patches and Updates

This section explains basic security implementations. With file encryption, files on the disk are protected using an encryption algorithm. Using private and public key encryption, packets traveling between two computers can be encrypted. SSL is a method for hiding the information a web browser and a web server send to each other. SSL is supported with a flash animation shown in Figure 5.7.

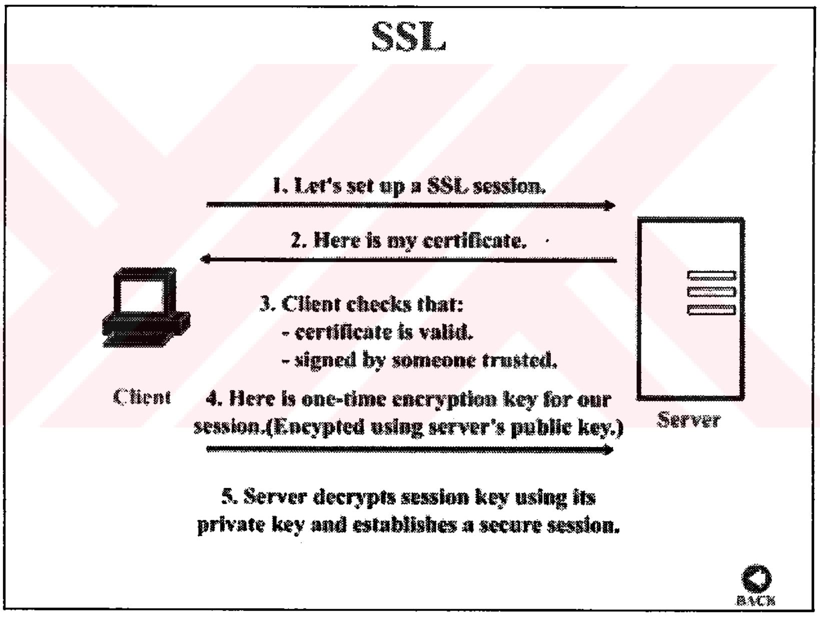

Figure 5.7. SSL in five steps

Next subsection explains IPSec. The IP Security protocol was developed for security of data as it is sent over a network. Fifth part is about firewalls. The concept that stands behind the firewall approach is to allow local users to enjoy full network services within their local network and some useful services provided by the Internet while controlling outsiders' access to the local network resources.

At the end, OS patches and updates are covered. In today's mission critical IT environments, reliability, availability, and service ability (RAS) are indispensable.

Software patches provide the means of performing software maintenance that, when handled properly, contribute to improved RAS, and thus the successful operation of the business.

5.8. Making the Course SCORM Compatible

This section explains how to make electronic content SCORM compatible. To make the course SCORM compatible, the electronic files should be associated with each other using an xml file named "imsmanifet.xml". In this section, the important parts of this file are introduced.

A manifest is a mandatory XML file that describes the components of a content package, much like a "packing slip". The manifest consists of the following sections:

i. Meta-data section describes the package as a whole
ii. Organizations section describes one or more hierarchical organizations of the content (content structure)
iii. Resources section references the actual resource and media files necessary for the

5.8.1. Meta-data Section

The term "meta-data" refers to data, actually information, about data. Course Meta-data is described in the imsmanifest.xml file.

The SCORM provides nine categories of meta-data for both assets and SCOs, each with several sub-categories. The nine categories are as follows (ADL, 2002):

i. General – information that describes the resource as a whole
ii. Lifecycle – features related to the history and current state of the resource and the individuals who have created the resource
iii. Meta-metadata – information about the meta-data record
iv. Technical – technical requirements and characteristics of the resource
v. Educational – educational characteristics of the resource

vi. Rights – intellectual property rights and conditions of use for the resource
vii. Relation – the relationship between this resource and other targeted resources
viii. Annotation – comments on the educational use of the resource and information on when and by whom the comments were created
ix. Classification – where this resource falls within a particular classification system.

Although meta-data are not required by SCORM, if the course is to be available for searching and extracting from a SCORM repository, then course meta-data should be included in the manifest file.

In the thesis, just three out of nine meta-data categories are used. These are "general", "classification" and "technical". Table 5.1 shows xml code for used categories.

Table 5.1. Used meta-data categories

```
- <imsmd:lom xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2">
+ <imsmd:general>
+ <imsmd:classification>
+ <imsmd:technical>
  </imsmd:lom>
```

In category "general", "title", "keyword" and "description" subcategories are used. Table 5.2 shows xml code for category "general".

Table 5.2. XML code for category "general"

```
- <imsmd:general>
 - <imsmd:title>
   <imsmd:langstring xml:lang="en">CMPE 150 Course Content</imsmd:langstring>
  </imsmd:title>
```

```
  <imsmd:language>English</imsmd:language>
- <imsmd:keyword>
   <imsmd:langstring xml:lang="en">Electronic Course Content</imsmd:langstring>
   <imsmd:langstring xml:lang="en">Flash Animation</imsmd:langstring>
   <imsmd:langstring xml:lang="en">Bogazici University</imsmd:langstring>
   <imsmd:langstring xml:lang="en">Computer Engineering</imsmd:langstring>
   <imsmd:langstring xml:lang="en">Introduction to Computing</imsmd:langstring>
   <imsmd:langstring xml:lang="en">Distance Learning</imsmd:langstring>
  </imsmd:keyword>
- <imsmd:description>
   <imsmd:langstring xml:lang="en">Following modules are included: IT Basics
Computer   Basics OS Basics Networking Basics Internet Basics WWW Basics Network
       Security</imsmd:langstring>
 </imsmd:description>
</imsmd:general>
```

In category “classification”, “purpose” subcategory is used. Table 5.3 shows xml code for category “classification”.

Table 5.3. XML code for category “classification”

```
- <imsmd:classification>
- <imsmd:purpose>
 - <imsmd:source>
    <imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
   </imsmd:source>
 - <imsmd:value>
```

```
   <imsmd:langstring xml:lang="x-none">Educational Course
Content</imsmd:langstring>
   </imsmd:value>
  </imsmd:purpose>
 </imsmd:classification>
```

In category "technical", "format" and "requirement" subcategories are used. Table 5.4 shows xml code for category "technical". There are two important technical requirements to follow the electronic, one is about browser type, and other is about flash plug-in. These two important requirements are specified in the meta-data section.

Table 5.4. XML code for category "technical"

```
- <imsmd:technical>
   <imsmd:format>text/html</imsmd:format>
 - <imsmd:requirement>
  - <imsmd:type>
   - <imsmd:source>
      <imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
     </imsmd:source>
   - <imsmd:value>
      <imsmd:langstring xml:lang="x-none">Browser</imsmd:langstring>
     </imsmd:value>
   </imsmd:type>
  - <imsmd:name>
   - <imsmd:source>
      <imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
```

```
</imsmd:source>
- <imsmd:value>
<imsmd:langstring xml:lang="x-none">Microsoft Internet
Explorer</imsmd:langstring>
</imsmd:value>
</imsmd:name>
<imsmd:minimumversion>4.01</imsmd:minimumversion>
</imsmd:requirement>
- <imsmd:requirement>
- <imsmd:type>
- <imsmd:source>
<imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
</imsmd:source>
- <imsmd:value>
<imsmd:langstring xml:lang="x-none">Plug-in</imsmd:langstring>
</imsmd:value>
</imsmd:type>
- <imsmd:name>
- <imsmd:source>
<imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
</imsmd:source>
- <imsmd:value>
<imsmd:langstring xml:lang="x-none">Macromedia Flash</imsmd:langstring>
</imsmd:value>
</imsmd:name>
```

```
<imsmd:minimumversion>5.0</imsmd:minimumversion>
</imsmd:requirement>
</imsmd:technical>
```

5.8.2. Organizations Section

The organizations component is used to provide structure to the content. Typically, this structure is provided in the form of a learning arrangement hierarchy. The IMS Content Packaging Specification does not bind the user to any particular structure. The organizations component provides the means to describe any number of different taxonomies that may be required.

Prepared course has a hierarchical structure which is shown in Figure 4.4 in section 4.2.1. There is a three-level hierarchy in the course content. Lets take chapter one. Name of chapter one is “IT Basics”, first second-level section is “Getting Started in IT” and first two subjects in this section are “Computer Systems and Programs” and “Computer Types”. To instruct to the LMS software this hierarchical structure, xml code shown in Table 5.5 is written.

Table 5.5. XML code written for organizations section

```
- <organizations default="MANIFEST01_ORG1">
 - <organization identifier="MANIFEST01_ORG1">


  - <item identifier="MANIFEST01_ITEM2"
identifierref="MANIFEST01_RESOURCE1" isvisible="1" parameters="Chapter 1">
     <title>IT Basics</title>
    - <item identifier="MANIFEST01_ITEM9" isvisible="1" parameters="">
       <title>Getting Started in IT</title>
     - <item identifier="MANIFEST01_ITEM10"
identifierref="MANIFEST01_RESOURCE3" isvisible="1" parameters="">
```

```
<title>Computer Systems and Programs</title>
</item>
- <item identifier="MANIFEST01_ITEM11"
identifierref="MANIFEST01_RESOURCE4" isvisible="1" parameters="">
<title>Computer Types</title>
</item>
.
.
.{other sections of chapter one}
.
.
</item>
</item>
.
.{hierarchical organizations of other six chapter}
.
</organization>
</organizations>
```

Three-level hierarchy of each chapter is constructed using xml with this methodology, and organizations part of the "imsmanifest.xml" is created.

5.8.3. Resources Section

The resources section can describe external resources, as well as the physical files that the package consists of. These files may be media files, text files, assessment objects or other pieces of data in electronic form. Conceptual groupings and relationships between files can be represented within the resources component. The resources are referred to at various points within the organizations component, which provides the structure for the resources.

There is a close relationship between organizations section and resources section. In Table 5.5, item name "MANIFEST01_ITEM2" references "MANIFEST01_ RESOURCE1" for its resources, or item named "MANIFEST01_ITEM11" references "MANIFEST01_ RESOURCE4" for its resources. Resources section for these two items is shown in Table 5.6.

Table 5.6. XML code written for resources section

```
- <resources>
  - <resource identifier="MANIFEST01_RESOURCE1" type="webcontent"
href="ch1\1_1\overview\content.htm">
     <file href="ch1\1_1\overview\content.htm" />
    </resource>
  - <resource identifier="MANIFEST01_RESOURCE2" type="webcontent"
href="ch2\2_1\overview\content.htm">
     <file href="ch2\2_1\overview\content.htm" />
    </resource>
        .
        . {resources for other items in this chapter}
        .
  - <resource identifier="MANIFEST01_RESOURCE4" type="webcontent"
href="ch1\1_1\1_1_2\index_2.htm">
     <file href="ch1\1_1\1_1_2\index_2.htm" />
     <file href="ch1\1_1\1_1_2\media.html" />
```

```
    <file href="ch1\1_1\1_1_2\1_1_2.swf" />

    <file href="ch1\1_1\1_1_2\content.htm" />

   </resource>

        .

        . {resources for other chapters}

        .

<resources>
```

5.8.4. SCORM Conformance Test

In order to make the course content SCORM compatible, an imsmanifest.xml file is written using xml. This file is tested with Microsoft LRN Viewer. There is no need to course index or back/next buttons to navigate through the course content.

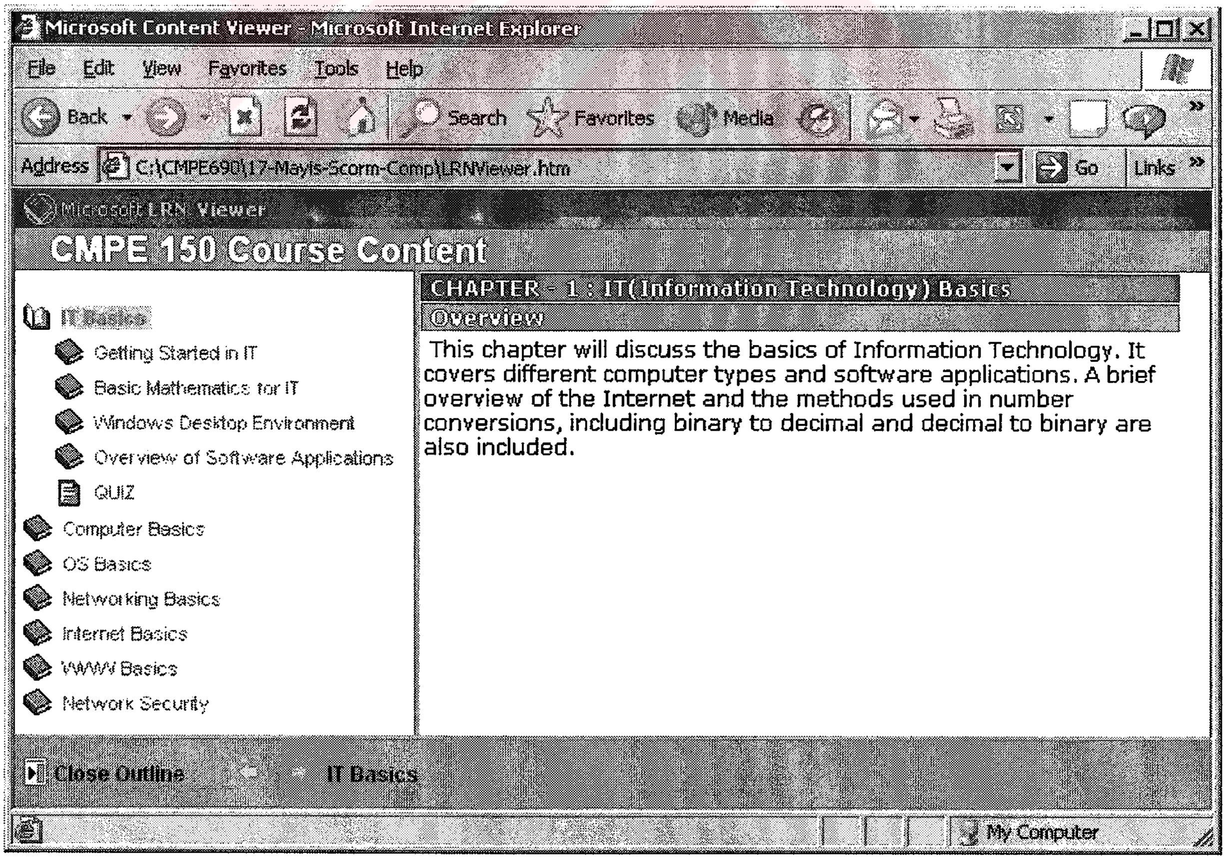

Figure 5.8. LMS conformance test

Relations are defined in the imsmanifest.xml file and LMS software displays the course content like defined in this file. Figure 5.8 shows the test result using Microsoft LRN Viewer. The user can navigate through the content using the index which is prepared by LMS software.

YK

6. CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

In this project, a distance learning system for the educational content of "Introduction to Computing" course is built. Distance learning is any learning that takes place with the instructor and student geographically remote from each other. DL has many advantages. Some of them are: time and place flexibility; potential to reach a global audience; easy updating of content, as well as archival capabilities.

The course content is constructed from the beginning to teach the computer engineering freshmen the basics of the IT world. Five steps are done for the thesis:

i. Seven subjects are chosen for the electronic content. These subjects are the basic ones for IT world.
ii. A web interface is designed to submit the course content. A functional, user-friendly web interface is created. Navigation is simple and easy to understand.
iii. Topics and subtopics of the seven subjects are decided and the content is created.
iv. Supporting flash animations are prepared.
v. SCORM compliant manifest file is written to state the structure of the electronic content to the LMSs.

Since the seven learning modules are always accessible to students via the WWW, they can review information and learn at their own place outside of class. So, this material does improve the effectiveness of learning.

At the end of the thesis, a framework has come out to transport the other departmental courses to the electronic world. The used methods and interfaces can also be applied to other CMPE courses.

6.2. Future Work

Several suggestions can be presented for future improvements for the thesis.

The development team of this project consists of only two engineers. For a successful learning project, social impact of the project should also be considered. Educators or psychologist can be consulted to improve the social part of the constructed learning system.

SCORM compatibility is implemented just in content package environment. The purpose is to explain the structure of the electronic content to an LMS. For a further improvement, SCORM Run Time Environment can be implemented. Run Time Environment API is a standardized way to send information back and forth between the learner and the LMS.

SCORM metadata is written just in manifest file, which means the written metadata can be used to get general information about the whole course content, not about all the components of the content. Metadata can be defined for each part of the electronic content, even for images, web pages or flash animations. To create a full reusable and searchable content, each electronic part should be defined with its own xml file which involves the metadata about the mentioned electronic part. The metadata part can be extended through the addition of xml files for each component.

REFERENCES

ADL, 2001, Sharable *Content object Reference Model (SCORM) Version 1.2*, http:// www.adlnet.org

ADL, 2002, *The SCORM Implementation Guide: A Step-by-Step Approach*, http:// www.adlnet.org

Adobe.com, 2004, *Adobe Photoshop Product Overview*, http://www.adobe.com/products/ photoshop/overview.html

Bleyle, J., 2001, *ActionScript Tutorial*, http://www.macromedia.com/support/flash/ action_scripts/actionscript_tutorial

Blue Star Learning, 2004, *What is FrontPage?*, http://www.bluestarlearning.com/ what_is_frontpage.cfm

Clements R., 2002, *Frame By Frame, Motion and Shape Tweening*, http://www.thepuma.co.uk

Cooper K., 2002, *Java vs. Flash*, Washington State University, http://www.math.wsu.edu/ kcooper/ flashplots/text3.html

Cyberlink, 2002, *Why SCORM?,* http://www.gocyberlink.com/english/products/ streamauthor/ articles/scorm.jsp

Drayman M., 2002, *Java VS Flash*, http://www.winneronline.com/articles/september2002/ javavsflash.htm

Ghiglione E., 2003, *Adaptive Learning using Learning Objects*, http://weg.ee.usyd.edu.au/ people/ernieg/thesis

Greenberg L., 2002, *LMS and LCMS: What's the Difference?,* http:// www.learningcircuits.org/2002/dec2002/greenberg.htm

Homan, S., 1997, "Distance Education Compared to Traditional Instruction: The Student's View", *International Journal of Instructional Media*, Vol. 24-I3, pp. 207-224.

JCA Solutions, 2002, *Online SCORM Course*, http://www.scorm.tamucc.edu/scorm/ home.html

Kerman P.,2003, *Teach Yourself Macromedia Flash MX 2004 in 24 Hours*, Sams.

Macromedia Livedocs, 2003, *Tweening Instances, Groups, and Type*, http://livedocs. macromedia.com/flash/mx2004/main/wwhelp/wwhimpl/common/html/wwhelp.htm? context= Flash_MX_2004_Documentation&file=09_mot17.htm

Macromedia, 2004, *Macromedia Flash MX 2004*, http://www.macromedia.com/ software/flash

Microsoft, 2001, *Microsoft FrontPage 2002 Tutorial*, http://www.microsoft.com/ downloads/details.aspx?FamilyID=6CF0ABC2-A6D8-43D1-B1D8-AC4BF751AA AC&dis playlang =en

Mischook, S., 2004, *What is Flash, When and Why to Use It?*, http://www.killersites.com/ articles/ articles_FlashUse.htm

Naps, T., J. Bergin, R. Jiménez-Peris, M. McNally, M. Patiño-Martínez, V. Proulx and J. Tarhio, 1997, "Using the WWW as the Delivery Mechanism for Interactive, Visualization-Based Instructional Modules", *In Supplemental Proceedings of the Conference on Integrating Technology into Computer Science Education: Working Group Reports and Supplemental Proceedings (ITiCSE-WGRSP '97)*, Association for Computing Machinery, pp. 13-26.

NetDimensions, 2002, *An Introduction to SCORM 1.2 Packaging*, http://netd5.netdimensions.com/nd/fresco/help/en/webhelp30/admin/white_papers/scormwhitepaper.htm

NetKontoret, 1999, *Flash Tutorial*, http://www.echoecho.com/flash.htm

Nichani M., 2001, *LCMS = LMS + CMS*, http://www.elearningpost.com/features/archives/001022.asp

Qu C. and Nejdl W., 2001, "Towards Interoperability and Reusability of Learning Resource: a SCORM Conformant Courseware for Computer Science Education", *The 2nd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2002),* Kazan, Tatarstan, Russia.

Rahm, D., and Reed, B. J., 1997, "Going Remote: The Use of Distance Learning, The World Wide Web, and the Internet in Graduate Programs of Public Affairs and Administration", *Public Productivity and Management Review*, Vol. 20-4, pp. 459-471.

Reinhardt R., 2001, *Flash 5 Bible*, John Wiley and Sons.

Schank, R., 1994, "Active Learning through Multimedia", *IEEE Multimedia*, Vol. 1, pp. 69-78.

SCORM CAM, 2001, *The SCORM Content Aggregation Model Version 1.2,* http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=837

SearchWebServices.com, 2001, *ActionScript*, http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci780181,00.html

SitePoint, 2003, *Flash Script - Shape Tweening in Flash*, http://www.sitepoint.com/ article/script-shape-tweening-flash?ct=1

Sol, S. and Tachibana, K., 1998, *Introduction to Web Design with Adobe Photoshop*, http://wdvl.internet.com/Authoring/Graphics/Tools/Photoshop/toc.html

Spigarelli B., 2004, *Instructional Design Article,* http://www.adlnet.org/index.cfm ?fuseaction=developer&pageview=viewarticle&ID=4

Sun, C. T., and Sunny, S. J., 2001, "Learning through Collaborative Design: A Learning Strategy on the Internet", *Proceedings of the 31st ASEE/IEEE Frontiers in Education Conference.*

Tsichritzis, D., 1999, "Reengineering the University", *Communications of the ACM* , Vol. 6, pp. 93-100.

Weil K., 2002, *Macromedia Flash MX Hands-On-Training*, Peachpit Press, ISBN: 0321112725.

Wetzel, C., P. Radtke and H. Stern, 1994, *Instructional Effectiveness of Video Media*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Verduin, J.R. and Clark, T.A, 1991, *Distance Education: The Foundations of Effective Practice*, San Francisco, CA: Jossey-Bass Publishers.

Yoakam M., 1996, *Distance Education: an Introduction*, http://www.digitalschool.net/edu/ DL_intro_mYoakam.html

REFERENCES NOT CITED

Armstrong G., *One Approach to Motivating Faculty to Use Multimedia*, http://www.thejournal.com/magazine/vault/A550.cfm, 1996.

Brajeshwar, *SCORM, What Is, What Isn't, Flash and Scorm and SCORM FAQ*, http://www.brajeshwar.com/archives/000155.php, 2003.

Code&data, *E-Learning : Learning Management*, http://www.code.com.au/elearning/management/lms.php, 2003.

Knowledgepresenter, *Knowledgepresenter Learning Management System*, http://www.knowledgepresenter.com/support/lmsmain.htm, 2004.

Mindflash, *Mindflash Learning Management System*, http://www.mindflash.com/pages/home.asp, 2003.

Sönmezer B. and Bingöl H., *E-Öğrenim İçin Scorm Uyumlu Elektronik İçerik*, BTİE 2004, METU, Ankara, 2004.