Developing Instructional Web Sites - A Software Engineering Approach

JONAS A. MONTILVA
University of Los Andes, Faculty of Engineering, Computer Science Department, Mérida, Venezuela 5101
E-mail: jonas@ing.ula.ve

BEATRIZ SANDIA
University of Los Andes, Academic Vice Rector's Office, Center for Interactive Distance Education, Mérida, Venezuela 5101
E-mail: bsandia@ing.ula.ve

JUDITH BARRIOS
University of Los Andes, Faculty of Engineering, Computer Science Department, Mérida, Venezuela 5101
E-mail: ijudith@ing.ula.ve

Full mailing address:

Corresponding author: Jonas Montilva
Universidad de Los Andes
Facultad de Ingenieria
Departamento de Computacion
Merida, 5101
VENEZUELA
Telephone: 58-74-403811
E-mail: jonas@ing.ula.ve
Fax number: 58-74-403879

Developing Instructional Web Sites - A Software Engineering Approach

Abstract

This paper focuses on the application of good practices to the development of instructional web sites. The development of a web site, to support a distance or campus-based online course, is a very complex process that involves many instructional and technical aspects, including the instructional purpose of the course, its content, its structure, its interface, and its teaching-learning activities. The use of a method is essential for dealing with the complexity of this process. We describe, in this paper, a software engineering method that helps course designers and teachers to develop high quality instructional web sites. The design of the method was based on the integration of well-known principles, concepts and process models borrowed from instructional design and object-oriented software engineering. The method addresses the managerial processes required to plan, organize, and control the project, as well as the instructional and technical activities involved in the development of high quality course sites.

Key words: Instructional Web Sites, Course Sites, Web-based Study Guides, Web-based Instruction, Distance Education.

Introduction

The World Wide Web (WWW) is currently the most common technological media used for developing and delivering online courses and instructional material. Many distance education courses are actually being offered through Internet by using the WWW. In campus-based education, WWW is also an important publishing medium for the development of online syllabus, study guides and other instructional materials.

Delivering an online course via the WWW involves the use of a special kind of web site, usually called *instructional web site*, *web course site* or *course site*, for brevity. A course site is a teaching-learning environment implemented and delivered through WWW to support an online course. It can be created and maintained using either generic web editors or more specialized course management software tools.

For creating course sites, many developers apply general-purpose methods, such as those used for developing conventional web pages and hypermedia applications (see, for example, Lynch and Horton (1999) and Conallen (2000)). The problem of using these methods is that they do not deal explicitly with those requirements that are proper to online courses. These methods are too general to be effectively applied to the design and development of course sites. They do not address those aspects that are specific to education, such as the course content and the teaching-learning process. Domain-specific methods are, therefore, needed to describe and guide effectively and efficiently the process of developing course sites.

Several methods and design principles, devoted exclusively to the development of online courses, have been published in the literature during the last five years (see, for example, Khan (1997), Hall (1999), McCormack & Jones (1998), O'Sullivan (1999), Pernici & Casati (1997), and Pallof & Pratt (1999)).

Hall (1999) and O'Sullivan (1999), in particular, focus on the definition of relevant issues and design principles for developing course sites, study guides, online syllabus and other kinds of online instructional material. Boling & Frick (1997) and Pernici & Casati (1997) concentrate exclusively on the design activities, which are part of the whole development process. But only a few of the cited authors describe the entire process of developing an online course from its planning to its delivery. Pallof & Pratt (1999), for example, discuss the whole process, but they concentrate only on the course syllabus. McCormak & Jones (1998) describe the process of developing course sites using a conventional approach known as the waterfall model. This approach has, however, several drawbacks, such as its failure to treat software as a problem solving process, its inability to iterate and evolve between development activities, and its tendency to consider software development as a manufacturing process rather than a creative process (Pfleeger, 1998).

An object-oriented software engineering approach may provide a more comprehensive view of the course site development process. By focusing on the objects of a course site, this approach may provide a more natural and complete definition of the development process that includes not only the design, but also other important development activities, such as the analysis of the course, the requirements specification, the implementation and the evaluation of the course site. Additional benefits are the following:

- a better structure and visibility of the process, which are essential for planning and controlling the project;
- a better quality of the instructional products; and
- a better managerial control of time, effort, and resources required to produce a course site.

We describe, in this paper, an object-oriented software engineering method for guiding instructors and course designers during the process of developing course sites to support online courses. We first define the general requirements that a method for developing course sites should meet. Then, we present a conceptual model that captures the main objects of a course site, as well as its functionality and technology. This model is essential for gaining an understanding of the purpose, general characteristics, and structure of a course site. It provides the conceptual framework needed to describe and use the method. The method structure is then analyzed in detail, followed by a brief description of its phases and steps. Finally, an evaluation of a course site developed using the method is given before the concluding remarks.

Requirements for course site development methods

Developing course sites to support web-based courses demands specific requirements that make this process very different from the development of any other type of WWW sites, such as corporate home pages and e-commerce sites. Some of the most important and general properties of a course site, that are not present in other type of web applications, are the following:

- the instructional objectives of the course,
- the structure and content of its study guide,
- the mode and media used for the interaction among instructors, students, and course content; and
- the teaching-learning process.

In addition to explicitly addressing these properties, a method for developing course sites should also comply with some general requirements. The following list is an extension of the requirements specified by Welsh (1997):

1. Systematic: The method should describe what to do and provide guidelines for how to do the activities required to produce the components of a course site.
2. Complete: It must cover the whole life cycle of a course site, which involves the typical phases of the software engineering process: requirement analysis and specification, design, implementation, testing, and delivery (Bruegge and Dutoit, 2000).
3. Adaptable: It must be modular and modifiable to allow its adaptability to different educational disciplines, pedagogical orientations, and development environments.
4. Technology independent: It must be independent of the software tools used to create, operate, and maintain the course site.
5. Quality Assurance: To ensure the high quality of a web study guide, the method must provide some form of quality assurance and the verification and validation of its intermediate and final products.

The purpose of the method is also a crucial aspect to be considered. A course site development method must guide the team of instructors and designers in the process of developing a course site. It must provide visibility, orientation and guidance to the team involved in this process.

A course site conceptual model

An essential element for designing any software development method is the conceptual model of the product to be developed.. This model describes the relevant concepts of the product, its components and their relationships. The purpose of this section is to describe the conceptual model that serves as a basis for designing and understanding the process model of the method, i.e., the set of activities required to develop a course site.

A course site is a virtual teaching-learning environment that can be seen and understood from, at least, four different perspectives or views: instructional, technological, structural, and functional.

*1) The instructional perspective***:** A course site can be seen as an instructional and communication medium that provides an environment to facilitate the teaching-learning process of a web-based course. It must support different kinds and modes of interactions. The interaction kinds define the communication lines between the participants and components of a course site. There are, at least, four kinds of interactions: instructors-content, instructors-students, students-content, and students-students. The interaction mode, on the other hand, determines the media (e.g., e-mail, videoconferences, bulletin board, discussion lists, and chats) that are used for establishing the communication between teachers and learners. The interaction mode can be asynchronous or synchronous. In the asynchronous mode, students interact with the content or the instructors at any time, without any space and time restrictions. Using e-mail and browsing the content of the study guide are two typical asynchronous activities. The synchronous mode, on the other hand, requires real-time interaction between two or more participants who should meet virtually at some pre-defined time. Chats and videoconferences are examples of synchronous activities.

2) *The technological perspective*: Using this perspective, the course site is seen as a collection of interlinked web pages stored in a web server and accessed from any client computer connected to the Internet. Using the course site requires a web browser on the client side. To initiate a session, the host page of course site must be invoked, through the web browser, using the identity or address of the site known as the uniform resource identifier (URI). A login name and password is usually required to protect the site from unauthorized access.

3) The structural perspective: This view focuses on the components of a course site. As Figure 1 shows, a course site is composed of a site management tool, a web study guide, and a set of interaction tools..

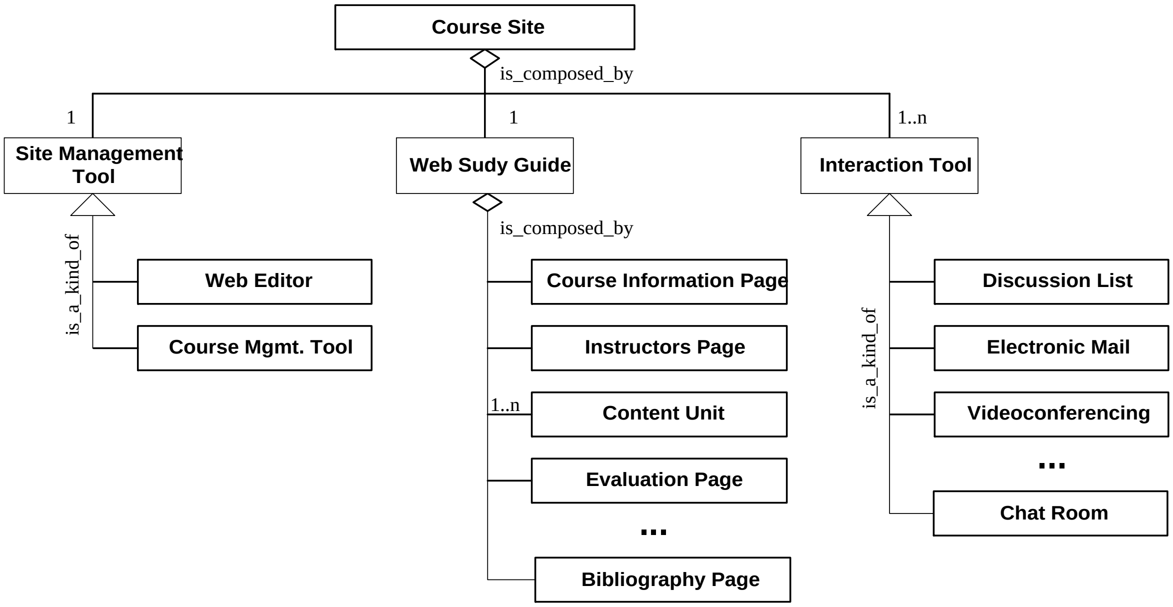

Figure 1. The structure of a course site

The site management tool is a web-based software system that provides the functionality needed to create, manage, update, and maintain a course site. It could be a web general-purpose tool (e.g., a web editor or a web integrated environment, such as Netscape Communicator™), or a more specialized web tool, such as a course management tool (e.g., Virtual-U (Harasim, 1999), WebCT™, CourseInfo™, JLE™, or TopClass™).

The set of interaction tools is a collection of media or collaborative technologies that facilitates the communication between instructors and students. Discussion boards, electronic mail, videoconference viewers,, and chat room tools are some of the most common media used by learning communities (DeSanctis, et al., 2001).

The most important component of a course site is, however, the web-based study guide (or w*eb study guide*, for short). The study guide is widely recognized as the central component or "foundation of every good distance education course" (Moore, 1997). It describes and integrates the structure, the content, the interaction and the media needed to support the teaching-learning process. The study guide is also seen as the main communication mechanism to support different types of interaction among students, instructors, and course content (Moore and Kearsley, 1996).

*4) The functional perspective***:** This perspective concentrates on the behavior of a course site, i.e., on the operations that are available to the developers, instructors, and students. The course developers must have available, through the site management tool, the operations needed to create, administer, and maintain the site. The instructors must be able to create and update the course content of the web study guide. Other operations that are commonly used by the instructor are the monitoring of students, online grading, and the publishing of online material, guidance information, and instructions (e.g., announcements). The functionality available to the students must include additional activities other than reading the web pages, such as writing, viewing, practicing and testing. The site must also provide the functionality needed to establish remote communication between students and instructors.

The course site conceptual model, as described in this section, highlights the most important elements that are present in any course site. This model provided the concepts needed to design the method described next.

The structure of the method

The method was created through the integration of principles and process models used in Object-Oriented Software Engineering, Hypermedia, and Instructional Design. The structure of the method was designed based on an object-oriented process model for developing business applications (Montilva, 2000). The Instructional Systems Design model, proposed by Dick and Carey (1985), was used a reference framework to outline the steps needed to deal with the development of instructional material. For dealing with the process of modeling, which is crucial in any software development method, we chose the Unified Modeling Language UML (Booch et al., 1998). This language is widely known and has been broadly used by the software industry and academic community since its standardization in 1997.

The proposed method is structured into two processes. The first of them, called the management process, is concerned with the activities that are required to manage the development of a course site. The second one, called the development process, is related to the instructional and technological activities that are required to produce and deliver a course site. As illustrated in Figure 2, the structure of the method covers the complete life cycle of a course site: from the analysis of the course to the delivery of the course site as a final product. It resembles the structure of a watch. The outer part of the cycle corresponds to the development process. It consists of five sequential phases that are executed in clockwise order. The management process is at the center of the cycle to control the progress of the development process.

The application of the method starts at the managerial process and continues through the execution of the five phases of the development process. The managerial process determines the flow of execution and the iteration among the development phases.

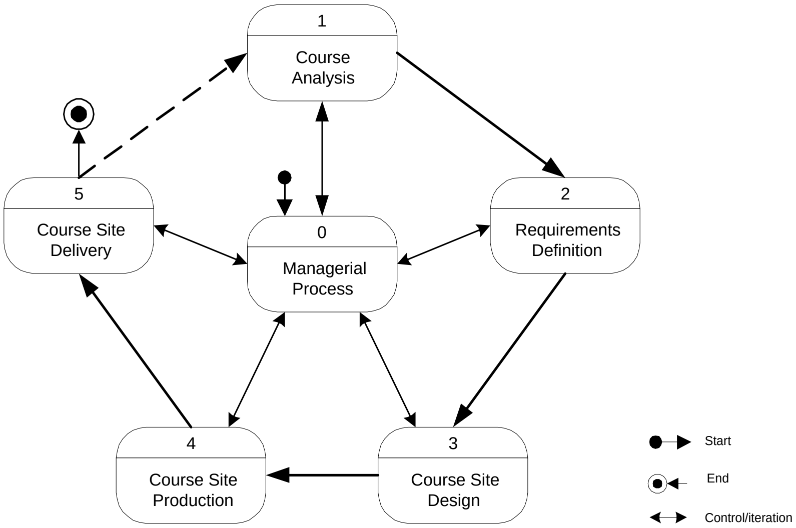

Figure 2. The structure of the method

The management process model

The managerial process is hierarchically structured into activities and sub-activities, as shown in Table 1. The activities produce one or more deliverables that are the results of applying selected managerial techniques and tools indicated in Table 1.

Table I. The management process

Activities	**Sub-activity**	**Techniques and tools**	**Deliverables**
Project Management	Project Planning and Control	Project management tool	Project plan Team structure Team responsibilities Control diagrams
	Team Organization & Staffing	Organizational structures, job description	
	Team Direction	Motivation, leadership, coordination	
Quality Assurance	Quality Planning	Quality Model	Quality Plan Product evaluation
	Verification & Validation	Technical reviews (e.g., Design Inspections and Walkthroughs) Web site design principles and standards	
Tools Selection & Acquisition	Selection of the course-creation software tools Tool acquisition	Software Evaluation and Selection	Site management tool
Training	Training the development team	Evaluation of needs and training selection	User´s guide
	Training the users	User’s guide production	

Project planning and control: Project planning is the first activity to be executed in the development of a project. It produces a schedule and a budget. The schedule defines the timing and order of the activities to be executed by the development team. The budget defines and distributes the cost of the project. Project control ensures that the project is executed according to the project plan and budget.

*Team organization & staffing***:** These activities are related to the structure and composition of the development team. A team integrated by a webmaster, one or more content experts (e.g., the course instructors), and one or more media specialists may be sufficient for developing most of the course sites. One of these members must be designated as the team leader. Obviously, the size and composition of the development team will depend on many factors, such as the scope of the course site, the budget assigned to the project, the time restrictions, and the availability of specialists. Moore and Kearsley (1996) discuss alternative team structures.

*Quality Assurance***:** The quality of a course site can be defined as the totality of features and characteristics of the course site that bear on its ability to satisfy its user's needs and requirements. The course site quality is ensured through the execution of two activities: quality planning and verification and validation..

Quality planning is a managerial activity that determines the quality features or characteristics of the course site, the verification and validation techniques to be applied to evaluate the products, the scheduling of verification and validation activities to be carried out, and the specific responsibilities of the quality team. The quality characteristics of a course site can be defined using quality models, such as the McCall's model or the Boehm's model, which establish the set of general characteristics that a software product must satisfy from the users and developers point of views (Pfleeger, 1998).

Verification and validation (V&V), on the other hand, are two complementary activities that ensure that the development products comply with its requirements and satisfy its user's needs. The verification and validation of a product is achieved by using technical reviews, such as design inspections and walkthroughs. A technical review is conducted at the end of each development phase by a quality team composed of a chairman, the product designer or developer, and a group of selected instructors and students. A technical review involves a meeting of the quality team to examine and check an intermediate or final product of a development phase. The purpose of the review is to determine the adequacy of the product with respect to its requirements (Ince, et al, 1993). The technical reviews ensure that the instructional products comply with its requirements, quality standards and well-established design principles, such as those indicated in (Lynch and Horton, 1999) and (Hall, 1999). These reviews also determines if the development team can advance to the next phase or must go back to a previous phase to incorporate new changes or correct errors detected in the products.

Tools selection and acquisition: It is concerned with the evaluation and acquisition of the course management software tool. O'Sullivan (1999) discusses some of these tools and provides an extensive list of tool comparison sites.

Training: Team training is performed through the development process to ensure that the development team has the skills and knowledge needed to use the method, the software tool and the development techniques. User training, on the other hand, is performed at the end of the development process to train students and instructors on the proper use of the course site.

The development process model

This process is structured into a set of five phases. Each phase is divided into steps and each step is, in turn, divided into tasks. The output of each phase is a set of verifiable products. The phases, steps and their products are summarized in Table II.

Table II. The development process and products

Phases	Steps	Products
Course Analysis	Analyzing the course domain	Domain and object model
	Analyzing the course objects	
Requirements	Defining the type of the course site	

Definition	Defining the functional requirements	Requirements model
	Defining the interaction requirements	
	Defining the development and operational requirements	
	Defining the quality attributes	
Course Site Design	Designing the structure of the web study guide	Structural model Content model Interface model Prototype
	Designing the content of the web study guide	
	Linking the learning activities to the interaction tools	
	Designing the user interface	
	Building a prototype of the web study guide	
Course Site Production	Producing the multimedia items and the study material	Web study guide
	Assembling the items and the study material into the prototype	
Course Site Delivery	Creating the course site	Course site
	Creating the user´s accounts	
	Training the users	

The relationship between management and development processes

The flow of control of the method and the relationship between management and development processes are shown in figure 3. The project starts at the project planning activity. The development phases are initiated once that the team is organized and the quality of the course site is planned. The products created by each development phase are reviewed by the V&V management activity. A great deal of iteration occurs between the development phases and the V&V activity. The development team moves from one development phase to the next development phase only when the quality team assures that the quality of the product is good enough.

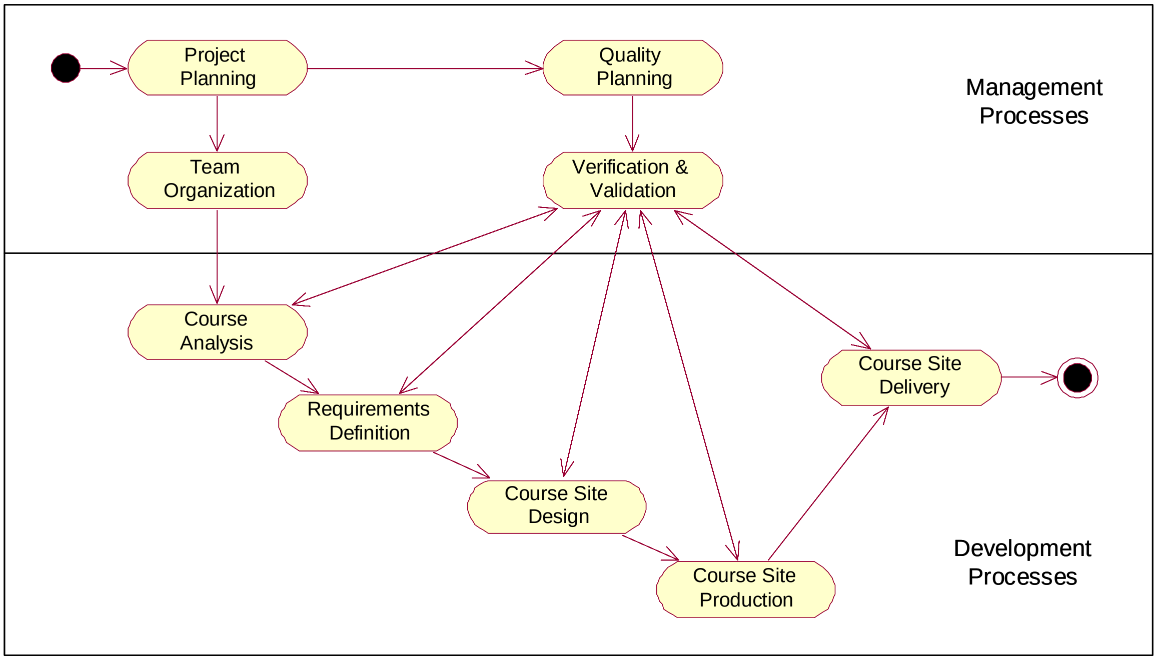

Figure 3. Flow of control of the method

The development process in detail

The development of a course site (http://www.centauro.ing.ula.ve/isoo) used by the authors to teach a software engineering course at the University of Los Andes in Venezuela is used, in this section, to explain in more detail and exemplify the steps, tasks and products of the development process.

Phase 1: Course Analysis

The initial development phase focuses on gaining a good understanding of the course that will be supported by the course site. Two aspects of a course that are critical to get this understanding are its domain and its objects.

Step 1.1: Analyzing the course domain. This step is concerned with the analysis of the context or environment of the course, i.e., the educational system or program of study to which the course belongs. Understanding this program is critical for a proper definition of objectives, needs and requirements of the course. Common tasks related to this step are the following:

- The identification of the program of study,
- The definition of its objectives, and
- The identification of pre-requisites and relationships that the course being developed maintains with other courses of the program.

Step 1.2: Analyzing the course objects. The main objects or entities involved in a course are the content, the student, the instructor, and the learning environment (Dean et al., 1996). The tasks of this step are concerned with the analysis of these objects, as explained next:

- *Analyze the content*: This involves the identification and analysis of the subject of the course, the search of relevant bibliography on the subject, the definition of the course content and its scope, the organization of this content in themes, and the definition of the learning objectives and goals of the course.
- *Analyze the students*: The main variables to be considered about the student are: the knowledge, skills and abilities to be learned; prior knowledge on the subject; skills required before taking the course; and the student profile.
- *Analyze the instructors*: Some of the most important variables or attributes to be considered about the instructors are the following: the subject-matter knowledge; their distance teaching experience and attitude; computer proficiency; their knowledge and experience on Internet services (WWW, FTP, E-Mail, News, etc.); and their pedagogical profile.
- *Analyze the learning environment*: It refers to the following aspects: the location of the students; the telecommunication technologies and hardware-software platform that they will have available at their learning sites; the social and physical environment; and the time availability of the students for taking the course.

Phase 2: Requirements Definition

This phase is concerned with definition of quality attributes, development restrictions, and functional requirements to be satisfied by the course site. Defining requirements are needed for two purposes: (1) to guide the development team about the most important features to be considered during the design phase; and (2) to ensure the quality of the course site through the verification and validation activities.

Step 2.1: Defining the type of the course site. Depending on the objectives of the course, the development team must define, in this step, the type of course site to be developed. There are, at least, four types of course sites: instructional, supplemental, dependent, and fully online (O'Sullivan, 1999). The informational and supplemental course sites are normally used as complementary aids to support the teaching process of a conventional or campus based-course. The dependent and fully online course sites, on the other hand, are essential to complete the course. They provide not only information but also communication to support the interaction between instructors, students and content. Fully online course sites are common in distance education.

Step 2.2: Defining the functional requirements. Based on the functional view of the conceptual model, this step defines the functional requirements of the course site, that is, the set of operations that the course site must provide to instructors and students. They can be divided into teaching operations and learning operations. Some of the most common teaching operations are the ability to create and update the course content of the web study guide, the monitoring of the students, the evaluation of students, and the publishing of online material, announcements and special instructions for the students. The learning operations include reading course information and online material, writing and submitting reports and assignments, viewing videoconferences, listening audio-conferences, group interaction, and online testing.

Step 2.3: Defining the interaction requirements. Interaction requirements are concerned with the ways the students and instructors communicate with each other through the course site and how they interact with the content provided by the web study guide. The following tasks are needed to identify and define these requirements:

- Define the kinds of interaction: The team must specify and describe here the kinds of interaction (i.e., instructor-content, student-content, student-instructor, and student-student) that the site will support.
- Define the interaction mode: Once the interaction is defined, the team must determine the media that are required to support each kind of interaction. E-mail, discussion lists, chats, videoconferences, student and group web pages, newsgroups, and file transfer are some of the most common types of media used in a course site.

Step 2.4: Defining the development and operational requirements. The design and production of a course site depends heavily upon the availability of limited resources, such as time, hardware, software, people and financial support. The purpose of this step is, therefore, to define the restrictions of time, cost, people, and computing resources under which the course site will be designed and produced in the remaining phases of the development process. The ongoing maintenance of the web site and the technical support needed by the instructors and the students must also be defined in this step.

Step 2.5: Defining the quality attributes. The quality of a course site is as important as its content. Achieving quality requires a commitment to well established web style rules and design criteria, such as those given in (Lynch and Horton, 1999), (Hall, 1999) and (McCormack and Jones, 1998). The quality of a course site is mainly determined by the quality of its study guide. These attributes can be grouped as follows:

- *Content attributes*: The content is at the heart of a study guide. Some of its most important quality attributes are: the scope of the content; the logical sequence and organization of the content; its coherence and completeness; the way of stimulating or motivating the student; the feedback on assignments; the methods used for evaluating the content; and the repetition and summary of the most important ideas. Furnell et al. (1999) discuss some of these requirements in detail.

- *Structural attributes*: The structure of a web study guide must be modular, visible, well balanced, easy to modify, and easy to navigate. A modular study guide helps the user to navigate smoothly through its content. Modularity also simplifies the process of updating and modifying the content of a study guide. It is achieved by dividing the structure of a web study guide into a set of hierarchical cohesive pages based on content (e.g., thematic chapters, sections and subsections). Visibility, on the other hand, is concerned with helping the student to see the study guide structure. Index pages and tables of contents can show clearly the structure of the study guide. Orientation is also needed to avoid getting lost in the hyperspace created by the study guide. Navigation aids such as navigation bars that link to "previous", "next", "home", "table of contents", and "index" pages should be included at the bottom or top of each page.

- *Interface attributes*: Students interact with the course content through the pages of the web study guide. These pages may contain many links and different types of multimedia items, including text, tables, still images or photographs, animations, video and audio clips. How these links and items are organized into the pages and visualized on the screen is a crucial factor for the successful design of the study guide. Specific attributes of the interface are the following: page length, background color and texture, design grids, size and resolution of graphics and images, and typographic design (Lynch and Horton, 1999).

Phase 2 concludes with the verification and validation of requirements and the selection of tools, as indicated by the management process (see Table I).

Phase 3: Course Site Design

Designing the course site is the most complex and time-consuming phase. The central component of the course site - the web study guide - is designed in this phase. The four perspectives or views of our course site conceptual model - instructional, technological, structural, and functional - are used here to define the main aspects to be considered during the design of the web study guide. These aspects are the content, the user-interface, the structure, and the interaction of the web study guide.

Step 3.1: Designing the structure of the web study guide. The structure of the study guide is usually hierarchical. A possible design of this structure is exemplified in Figure 1. At the top of the hierarchy is the host page or Web Study Guide main page, i.e., the first page that is displayed on a client computer when users connect to the course site. This page must present the title of the course, the name of the department or program of study, and the institution name and logo. This page must also include an index or framework of links to establish the navigation to the pages of the lower level of the hierarchy, as shown by the UML statechart diagram given in Figure 3.

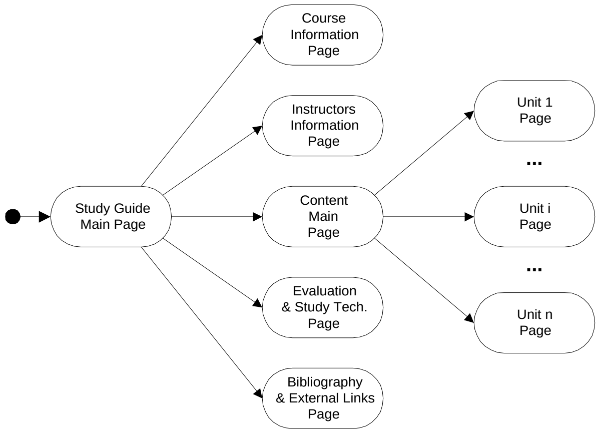

Figure 3. Navigational graph of a typical web study guide

The Course Information Page contains the course syllabus. It describes the learning objectives; gives an overview of the course content; and presents the schedule of classes. The Content Main Page describes the course content in detail and provides a link to each of the lessons or content units that conform the course. The rest of the pages are self-explanatory.

Step 3.2: Designing the content of the web study guide. The course content must be divided into a sequence of units, lessons or themes based on pre-defined criteria, such as the conventional structure of the subject, as exhibited by the main textbooks used to support the course. The course content is then distributed over a set of pages, called *unit pages*, which are accessed through the Content Page, as shown in Figure 3. Each unit page contains a unit or lesson that covers a specific content. For each unit page, the team must perform the following tasks:

- *Define the structure of the unit*: A unit may be structured into a set of sections, including the learning objectives (i.e., what the student must know at the end of the lesson); the specific content (i.e., what topics will be covered by the lesson); the learning activities (i.e., what the student must do to achieve the learning objectives); and the assessments (i.e., how the lesson will be evaluated). Figure 4 models the structure of a unit page.
- *Design the presentation of the unit:* The information of a unit page may be presented in many different ways. Lists and tabular organizations are two common ways of presenting the sections of a unit. Table 2 shows an example.

Table 2. Tabular organization of a unit page

Unit Number/ & Schedule	**Learning Objectives**	**Topics**	**Learning Activities**	**Evaluation**
5 Duration: 1 week From: 03/18/2002 To: 03/22/2002	1. To familiarize with well-known software process models. 2. To learn how to select the best model for a given project.	Software Process Models: 1. The Waterfall Model. 2. The Prototype Model. 3. The Object-Oriented Models.	Reading assignments: • Textbook : Read Chapter 12. • Read Unit 5: Web Material. Written Assignment: • Comparison of Models (see Assignment Description)	• Written assignment #5 (due on 03/22/2002) • Quiz #5: (03/27/2002)

Step 3.3: Linking the learning activities to the interaction tools. Some of the learning activities, specified in the previous step, involve the use of media or interaction tools. Appropriate links must,

therefore, be established from the learning activity section, in each unit page, to the media tools that will be used by the student. The statechart diagram in Figure 4 illustrates the associations between a unit page and the interaction tools required by the learning activities.

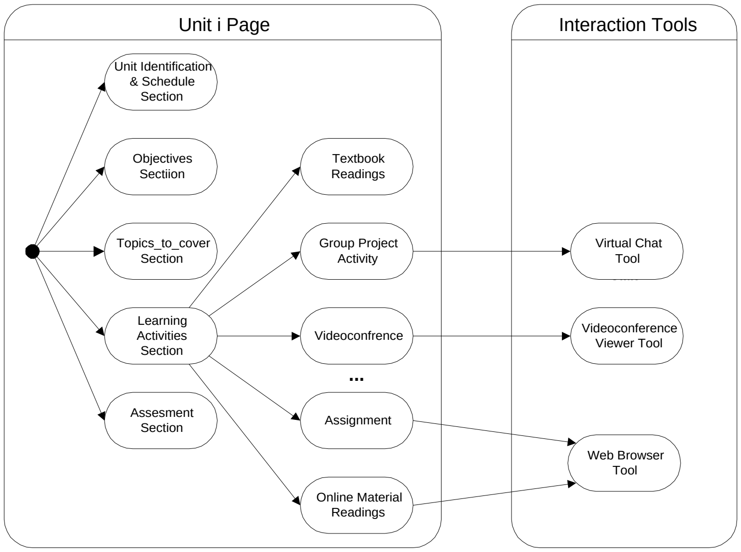

Figure 4. Structure of a unit and its association with the interaction tools

Step 3.4: Designing the user interface. The last aspect of the web study guide to be designed is its user interface. The purpose of this step is to define the aesthetics and visual characteristics of the web study guide. These details may be defined through the following tasks:

- *Design the visual structure and the navigation capability of the web pages:* A web page is composed of a set of multimedia items or data (e.g., text, graphics, still images, animation, and video), and a set of buttons that allows the user to navigate through the web study guide. The team must design the visual structure or layout of each one of the different types of web pages that compose the study guide. A page layout indicates the location of items and buttons on the rectangular area or space occupied by the page.

- *Design the multimedia items and buttons***:** The purpose of this task is to design in detail all the multimedia items and buttons to be included in the different pages of the web study guide. The content, art and visual details of each multimedia item and type of button should be outlined depending on its type. For instance, in the design of a video item, the development team must specify its content, size, duration and resolution.

Step 3.5: Building a prototype of the Web study guide. A prototype is a skeleton or framework that implements the structure of the study guide, which consists of pages, links and navigational aids. The purpose of the prototype is dual. Firstly, it is used for verifying and validating the design. The instructor and a selected group of students interact with the prototype to verify its design and validate its usability. Secondly, it is used as a framework for assembling the different items that are produced in Phase 4. The prototype will evolve into a finished product through the assembling of the multimedia items and its verification and validation, as explained in the next section.

The prototype must be constructed using the site management tool chosen by the Tool Selection management process. Building the prototype involves two tasks:

- *Create the structure of the prototype:* If the site management tool is a generic one, the structure of the prototype may be created just as it was designed in the steps 3.1 – 3.3. However, if a specialized course management tool is used, a mapping must be done between the designed structure of the study guide and the structure used by the tool. The pages and sections of the study guide structure must conform to the structure imposed by the course management tool.

- *Place the textual content into the prototype:* Most of the textual information associated with each page of the web study guide can be placed in the prototype using the site management tool.

Once the prototype is created, it is used by selected groups of students to verify and validate the design of the web study guide, as defined by the quality assurance activity of the management process.

Phase 4: Course Site Production

The production phase is concerned with the elaboration of the items and the study material that were designed in Phase 3, as well as their assembling into the prototype. This phase involves the following steps:

Step 4.1: Producing the multimedia items and the study material. This step may require the assistance or participation of specialists in audio-visual technologies and content. Producing a multimedia item, such as an animation or video clip, is a very complex and creative process for which special skills and abilities are usually required. Image, audio, animation and video items require special processing, such as digitalization, editing, storage, and format conversion. Multimedia experts may be needed for producing these types of information items. The study material that will be associated with the unit pages must also be produced during this step. Content experts or instructors must write the textual content of the study material and work with multimedia experts to produce the items required by the study material.

Step 4.2: Assembling the items and the study material into the prototype. Once the multimedia items and study material are produced, they must be assembled into the prototype using the linking capabilities of the site management tool.

Step 4.3: Integrating the components of the course site. The prototype can now evolve into a course site by integrating or linking the interaction tools and publishing the web study guide into a web server. This step depends on the site management tool. If a generic web editor is used, the development team must write the applications (e.g., scripts and applets) needed to link the interaction tools to the unit pages. This process is transparent when a course management tool is used, because it includes the facilities needed to integrate web pages and interaction tools.

Step 4.4: Validating the course site. As soon as the prototype evolves into a course site, it is tested again by a group of students to complete the validation process. The course content, the user-interface, the course site structure and the interaction media are evaluated by the students to determine if they comply with the requirements established during the execution of Phase 2. The results of this process are used for adjusting and refining the course site before its delivery.

Phase 5: Course Site Delivery

Once the course site has been validated, it is ready to be delivered and enter into operation. Some of the activities that must be conducted, in this final step, are the following:

- Producing the user's guide,
- Training the users, and
- Creating the student's accounts for using the course site.

This step signals the end of the development process and indicates the beginning of the operation and maintenance stage of the course site.

Evaluating the method

Based on the work of Jarke (1990) and Barrios (2001), a method evaluation framework was designed and used to compare and evaluate our method. This framework uses method engineering principles to establish the features and characteristics of the methods to be evaluated. Some of these principles are the following:

- A method should have a purpose that specifies the domain, the kind of problems to be solved, the conditions and restrictions to use it.
- A method must be composed by two essential and complementary models: a product model and a process model. The *product model* describes the intermediate and final products that are built using the method. The *process model* prescribes the actions that guide the team to develop the product.

The evaluation framework is composed of four complementary views of a method: domain view, usage view, product view, and process view. Each view is characterized by a set of facets along with their corresponding attributes. The values related to the attributes are assigned by the method evaluator during the evaluation process. The facets and attributes were adapted to characterize methods devoted to build up instructional web sites. The framework highlights the strengths and weaknesses of the evaluated methods.

The **Domain View** is related to the application domain of the method being evaluated. The domain view is described by two facets: the *scope* facet and the *instructional* facet. The scope facet allows the evaluator to appraise the method according to their coverage of technical, instructional and management features, and its application area. The *instructional* facet is concerned with the educational features of the web sites developed by the method. It includes the educational level, the instructional modality, the type of course to be supported by, and the dependence of the method on a particular approach. The values of each attribute are described next.

Scope:	Coverage: {Technical (T), Instructional (I), Management (M) }
	Application Area: {Education (E), Business (B), Other (O)}
Instructional:	Educational Level: {High School (H), Undergraduate (U), Postgraduate (P)}
	Modality: {Distance (D), Campus Based (C), Training (T)}
	Course Type: {Instructional (I), Supplemental (S), Dependent (D), Fully Online (F)}
	Approach: {Dependent (D), Independent (I)}

Table X. Domain View comparison table

Facet	Attribute/Values	Our Method	Lee and Owens' Method	Schweizer's Method
Scope	Coverage			
	Application Area			
Instructional	Education Level			
	Modality			
	Course Type			
	Approach			

The **Usage View** is concerned with the usability properties of the method. These properties are captured by the facets: applicability, and user level. The *applicability* facet evaluates the phases of the web site life cycle being addressed by the method and the main use of the method. The facet *user type* specifies whom the method is addressed to.

Applicability: Life-Cycle: {Analysis, Design, Development, Maintenance)
Main Use: {Academic, Commercial)
User Type: User Type: {Instructor, Specialist, Web Master, Web Designer)

Table X. Usage View comparison table

Facet	**Attribute/Values**	**Our Method**	**Lee and Owens' Method**	**Schweizer's Method**
Applicability	Life-cycle			
	Main Use			
User level	User			
Web Site Quality Control	Coverage			

The **Product View** is interested in the representation of the instructional web site that result from method application. The facets characterizing it are description, structure and content of the instructional web site. The *description* concerns the background concepts, the notation used, and the form by which concepts are related. The *structure* facet is related to the components of the product, its dependencies, and its adaptation capability to particular context uses. The *content* facet include attributes that allows to characterize products according to the concepts that can be took into account and the detail level (granularity) that can be used to represent them.

Product Representaion: Model: {Explicit (E), Implicit (I), Nonexistent (N)}
Model Description: Orientation: (object, event, functional, resources flow)
Notation: (formal, semi-formal, informal)

Model Structure: Components: (single, composed)
Dependency: (partial, complete)
Adaptability: (yes, no)
Contents: Concepts: (technical, instructional, ergonomic)
Granularity: (macro, micro)

Table X. Product View comparison table

Facet	**Attribute/Values**	**Our Method**	**Lee and Owens' Method**	**Schweizer's Method**

Description	Orientation			
	Notation			
	Form			
Structure	Components			
	Dependency			
	Adaptability			
Contents	Concepts			
	Granularity			

The **Process View** concerns the process of constructing the products specified in the product view. In our case, this view includes the process of building an instructional web site. It has been characterized by the facets Paradigm, Approach, Tool Support, Guidelines, User Participation, Process Execution and Process Type.

The facet *Paradigm* concerns the orientation of the process model defined to build up instructional web sites. The facet *Approach* permit to characterize process models according to their capacity to define development processes ad hoc or systematic. The facet *Tool Support* concerns the use of complementary tools during the development process. The facet *Guidelines* contains the attributes related to the way user is guided while the development process takes place. It has the associated attributes Nature, that allows characterizing the flexibility of guidelines, Granularity, that permits to determine the specificity level of guidelines, and Contingence which is associated to the guidelines capacity of adaptation to modeling situations.

The facet *User Participation* contains attributes that let to characterize process models according to the degree of user involvement in the development process, the way of organizing the project team, and team minimum size. The facet *Process Execution* characterizes the way of execution of the development process. The facet *Process Type* allows us to distinguish the different types of processes that participate in the process of building an instructional web site and that are covered by a process model.

Paradigm:	Orientation: (activity, product, decision, context)
	Approach: (ad hoc, systematic)
Tool Support:	Tool use: (prescribed, guided, independent)

Tool type: (general purpose, specialized)

Guidelines: Nature: (flexible, rigid)

Granularity: (macro, micro)

Contingence (yes, no)

User Participation: User involvement: (none, partial, completely)

Team organization: (interdisciplinary, specialist)

Team size (minimum)

Process Execution: Execution mode: (sequential, no-sequential, iterative)

Process Type: Development: (Analysis, Design, Implementation, Evaluation)

Planning & Control: (Risks, Contingence, Project, Records, Problems)

Quality Assurance: (Plan, Metrics, Quality management, Improve needs)

Post-development: (Installation, Operation, Maintenance, Refinement, Retirement)

Configuration Mgmt: (Plan, Identification, Control, Accounting)

Table X. Process View comparison table

Facet	**Attribute/Values**	**Our Method**	**Lee and Owens' Method**	**Schweizer's Method**
Paradigm	Orientation			
Approach	Approach			
Tool support	Tool usage			
	Type of tool			
Guidelines	Nature			
	Granularity			
	Contingency			
User participation	User's involvement			
	Team organization			
	Minimum team size			
Process execution	Execution mode			
Process types	Development			
	Planning & Control			
	Quality Assurance			
	Post-development			
	Configuration Mgmt			

Discussion

Developing a course site to support an online course is not a simple process. Many features, such as the instructional, structural, functional, and aesthetical properties of the site, must be considered. The method applied to develop the course site is also crucial, because it allows the development team to deal with the complexity of the process. The method, presented in this paper, is complete, modular, adaptable, technology independent, and simple enough to provide insight into the complexity of the course site development process. Its emphasis on quality assurance and its ability to integrate instructional, managerial and technical activities into a single methodological framework are also two important features of the method.

Our method differs from those described by Boling and Frick (1997), Pernici and Casati (1997), and Pallof and Pratt (1999) in at least two important aspects. Firstly, our method is more complete. It covers the whole life cycle of a course site and describes not only instructional and technical activities, but also the managerial ones. Secondly, our method uses the object-oriented approach, which has proved to increase the productivity of development teams, as well as the quality and adaptability of the products (Taylor, 1999). Our method is similar to one described by McCormack and Jones (1998). The main differences are based on the scope and approach used by the methods. Their method is more comprehensive than ours in the sense that it describes in more detail the development phases and includes a maintenance phase. Our method, however, gives more emphasis to the managerial activities and quality assurance, which are essential to develop efficiently and effectively good quality course sites.

Acknowledgements

This research has been sponsored by CONICIT - the Venezuelan national research council - under project No. G-97000823. The authors thank the anonymous reviewers for insightful comments. The first author is also grateful to the Software Testing Center of the Department of Computer Science & Engineering of the University of South Florida at Tampa, which provided a nice environment for completing the research.

References

Lynch, P. J. and Horton, S. (1999) Web Style Guide: Basic Design Principles for Creating Web Sites. Yale University Press, New Haven.

Conallen J. (2000) Building Web Applications with UML. Addison-Wesley, Reading.

Khan, B. H. (1997). Web-Based Instruction. Educational Technology Publications, Englewood Cliffs.

Hall, R. H. (1999) Instructional Web Site Design Principles: A Literature Review and Synthesis. The Virtual University Journal, 2, 1--12.

O'Sullivan, M. F. (1999) Worlds within Which We Teach: Issues for Designing World Wide Web Course Material. Technical Communication Quarterly, 8(1), 61--72.

Boling, E. and Frick, T. W. (1997) Holistic Rapid Prototyping for Web Design: Early Usability Testing is Essential. In Web-Based Instruction, B.H. Khan (ed), Educational Technology Publications, Englewood Cliffs, pp. 319--328.

Pernici, B. and Casati, F. (1997) The Design of Distance Education Applications Based on the World Wide Web. In Web-Based Instruction, B.H. Khan (ed), Educational Technology Publications, Englewood Cliffs, pp. 245--254.

Pallof, R.M. and Pratt, K. (1999) Building Learning Communities in Cyberspace. Jossey-Bass Publishers, San Francisco.

McCormack, C. and Jones, D. (1998) Building a Web-Based Education System. Wiley, New York.

Pfleeger, S. L. (1998) Software Engineering: Theory and Practice. Prentice Hall, New Jersey.

Bruegge, B. and Dutoit, A. H. (2000) Object-Oriented Software Engineering. Prentice Hall, New Jersey.

Welsh, T. M. (1997) An Event-Oriented Design Model for Web-Based Instruction. In Web-Based Instruction, B.H. Khan (ed), Educational Technology Publications, Englewood Cliffs, pp. 159--165.

Harasim, L. (1999) A Framework for Online Learning: The Virtual-U. IEEE Computer, 32 (9), 44--49.

Moore, M. G. (1997) The Study Guide: Foundation of the Course. Editorial note. The American Journal of Distance Education, 11(2), 1--2.

Moore, M.G. and Kearsley, G. (1996) Distance Education: A Systems View. Wadsworth Publishing Company, Belmont.

Booch, G., Jacobson, I. and Rumbaugh, J. (1998) The Unified Modeling Language User Guide. Addison Wesley, Reading.

Montilva, J.A. (2000) The Watch Process Model for Developing Business Software in Small and Mid-size Organizations. In Proceedings of the IV World Multiconference on Systemics, Cybernetics, and Informatics. Edited by the International Institute of Informatics and Systemics. Orlando, July, (1).

Dick, W. and Carey, L. (1985) The Systematic Design of Instruction. Scott, Foresman & Co, Glenview.

Dean, R., Biner, P. and Coenen, M. (1996) Distance Education Effectiveness: A System Approach to Assessing the Effectiveness of Distance Education. Education at Distance Journal, 10(3), J1--J4.

Furnell, S. M., Evans, M.P., Phippen, A.D. and Abu-Rgheff, M.N. (1999) Online Distance Learning: Expectations, Requirements and Barriers. Virtual University Journal, 2, 34--48.

Taylor, D. A. (1999) The Keys to Object Technology. In Handbook of Object Technology. S. Zamir (ed), CRC Press, Boca Raton, pp. 1.1 -- 1.12.