An Incremental, Spiral, Topic-Driven Approach

-- Designing labs and projects for a Web technology course

Jigang Liu
Department of Computer Science
Metropolitan State University
St. Paul, MN 55106
Jigang.liu@metrostate.edu

Kai Qian
Department of Computer Science
Southern Polytechnic State University
Marietta, GA 30060
kqian@spsu.edu

Abstract

How to teach Web technology through a single course is always a challenge for computer science educators because not only does Web technology update constantly, but also it covers such a wide variety of topics. In this paper, we will share our experience in teaching Web technology through a single course, particularly to discuss how a new incremental, spiral, and topic-driven approach helps students grasping sufficient-enough knowledge in Web technology so that they can establish a sophisticated-enough web-based application through a single course

Keywords

Web Technology, Computer Science Education, Software Development.

1.Background

In the ACM Computing Curricula 2001, Web technology is listed as Net-Centric 5 (elective), which gives us less than a single course class time to cover the subject. However, since Web technology involves a broad range of rapidly changing topics, such as programming languages, system environments, application tools, and design and implementation techniques, it is a great challenge for instructors to teach it through a single course. As a result, some schools emphasize only on teaching how to design Web applications and let students to figure out how to implement the technology [7]. Some schools are more interested in teaching Java technology so that the language issues become the focal point of the classes [1, 3]. The key issue discussed in [8] is how to select and incorporate leading-edge Web technologies into a Web programming course.

In order to cover as much topics as possible, some schools use a two-course sequence to introduce Web technology to their students [6]. Responding to the demand on the market, some schools even make their effort in designing a curriculum in Web technology [2].

Since the CC 2001 does not provide us with more than one course time to cover Web technology, it is very challenge to design a single course covering key components of Web technology. In next section, we discuss the new approach we have used in teaching a Web technology course. Then the lab/project design issues are presented in section 3. Implementation and teaching strategies are covered in section 4. Finally, conclusions are provided in section 5.

2. A New Approach

The approach proposed in [4] is different from ones we discussed previously. It is focusing on the functions a Web application should provide rather than languages. By using this approach, instructors can just teach critical topics in Web technology so that students can create a non-trivial three-tier web-based application through a single course. This approach provides us a new framework for teaching Web technology in a more effective way.

To support the new framework, we have applied the learning philosophy introduced in [5] in designing and implementing the labs and projects. To help students understand what is “learn how to learn,” we teach them how to look at the big picture rather than a particular cutting-edge technology first and then show them how to find out what is needed and where they can get it. In other words, we make students understand what to learn, know how to learn, and see what have learned. What to learn, how to learn, and what have learned compose of a learning cycle which is a foundation of our approach in designing the labs and projects for a Web technology course.

This new approach is spiral because students will repeat the learning cycle three times based on three projects, each of which includes client side and server side components and self-standing. It is also an incremental approach because each following project will add more complexity and functionality to their previous one. The new approach is also topic-driven because for each of three projects there is a main topic with several sub-topics to learn.

3. Design and Implementation Issues

In this section, we first present our design philosophy in general, and then discuss the lab design principles and project design strategies.

3.1 Design Philosophy and Consideration

Since Web technology has such a broad range of topics, the selection of the topics to be taught in class has been a very challenging task. On the other hand, to be able to create a non-trivial web-based application through a single course, students do not have time to learn everything from the scratch. Coping with the problem, an effective teaching framework was introduced in [4], where the course is constructed into three major components, namely lectures, labs, and projects.

Lectures are mainly utilized to introduce some of the abstract concepts and principles in Web technology as well as in the design and implementation of web-based applications. To stimulate students' learning and help them better understand the concepts, we designed labs so that students could feel and see the relationship between the abstract concepts and their corresponding outcome. After students take the lectures and complete labs, they should be able to implement the carefully designed projects. Through the implementation of these projects, what they have learned could be well evaluated and properly assessed.

Labs serve as a bridge between lectures and projects. Each of the labs is designed to meet a specific target. Students experience what they have been taught in the lectures through these labs. In addition to playing a role of the connection between thinking and understanding, labs also work as fundamental building blocks for students to create projects.

Anticipated goals are incrementally embedded in the functionality of the projects so that students can gradually grasp Web technology through these projects. Each project aims a particular target based on the knowledge students currently have just learned. At the meantime, each project provides a potential space for students to grow, which leads to the next project. .

3.2 Principles and Strategies in Lab Design

The labs are designed with well thought intention that is to assist students to experience and comprehend what they have learned in the lectures. Students then can manipulate these labs as the building locks in designing and implementing their projects.

Learning is a progressive process. To find out where should be the starting point for your students to learn or for the instructors to teach is the first challenge in designing labs. If the starting point is too lower, students would be bored. If the starting point is beyond the current knowledge of students, they will have troubles to connect what they have learned from class and what they do in labs. Although one can determine the background of students by setting proper prerequisites, the other way to find out the starting point is to give students a background check test in the first class. As soon as the staring point is determined, the three-difficulty-level labs can be properly designed.

Not only helping to keep students interest, short labs also assist students to complete short learning cycles on various subjects. Since a lab focus on a single language issue, students can easily see what a lab does, understand why it does, and know how it can be changed. As a result, students can learn more application building blocks through short labs and have also more flexibility in implementing projects.

If a longer lab is inevitable, showing intermediate results can have the same effect as short labs do. Due to the complexity of Web programming (issues on client-site programming, server-side scripting, web browsers, database management systems, and web servers), a lab on a particular Web application building block, such as shopping cart, could be much longer than other labs. Therefore, allowing a longer program to show some intermediate results will help students to check if they are on the right track.

The principles and strategies in the lab design are summarized in table 1 as shown below.

Table 1: Principles and Strategies in Lab Design

▪ A lab should be cohesive and present a single web technology topic in three levels of complexity, which are

simple, intermediate, and complex so that students can experience the foundation, feel the progress, and understand the principles through the evolution of the labs.

- A lab should be short enough so that most students can complete it within 20 minutes. Since students are facing numerous new concepts and technologies, it is a key to allow them to complete a learning cycle for a particular topic within a short period of time so that they can stay focus and be ready for the next lab which maybe on a complete different subject.
- A lab should be able to show some intermediate results every five to seven minutes during the time so that students could find out whether they are on the right track. At the same time, those intermediate results could also keep students' interest.

To illustrate how to apply the principles and strategies in design labs with three-difficulty-levels, a lab used to introduce Javascript with Events is provided as follow. The lab consists of three examples, which represent levels of difficulty, simple, intermediate, and difficult. Table 2 shows how a button can be programmed to process an event of "click."

Table 2 Button handling in Javascript

```
<html>
<head> <title> myEvent.htm </title>
<script language="JavaScript">
function dontclickonme() {
  alert("I told you not to click on me!");
}
</script>
</head>
<body bgcolor="blue">
<h2> An Event Example on JavaScript </h2>
<form>
<input tpye="button" value="Don't Click Me"
onClick="dontclickonme()">
</form>
</body>
</html>
```

How to deal with events on string processing is presented in Table 3. The example shown here is just a starting point that can help students to see how to get words out of a string. From there, students can expand the example to an application building block to check an email address or a phone number.

If one wants to know how to handle an event from the server-side, Table 4 provides a simple example. Through this example, students see how the connection between a client-side javascript program and a server-side Perl program can be established. With this connection, students can add more functionality to either the client-side or the server-side programs.

Table 3 String processing in Javascript

```
<html>
<head> <title> myString.htm </title>
<script language="JavaScript">
var strItems = new Array();
var strInput = "";
function strSplit(myStr) {
  document.myForm.items.value = "";
  strItems = myStr.split(":");
  for ( var j = 0 ; j < strItems.length ;
j++ ) {
    document.myForm.items.value            =
document.myForm.items.value + strItems[j] +
"\n";
  }
}
</script>
</head><body>
<h2>  Please  enter  items  separated  with
colons, then press <i>tab</i> key.
</h2>
<form name="myForm">
<input type="text" name="myInput" size="60"
onChange="strSplit(this.value)">
<br>
<h2> The items entered are listed as follows
</h2>
<textarea name="items" rows=5 cols=20>
</textarea>
</form></body></html>
```

Table 4 Handling events from the server-side

```
<html>
<head> <title> mySubmit.html </title>
<script Language="JavaScript">
function myCheck(myForm) {
  if (confirm("Are you ready to submit this
form?")) {
    myForm.submit();
  }
}
</script>
</head>
<body>
<form action="/cgi-bin/ics325??/first.pl">
<H3>
Please  enter  your  course  number  and
score:<br>
Course: <input type="text" length=20>
<br>
Score  :  <input  type="text"  length=10
onChange="myCheck(this.form)">
<br>
</H3>
</form></body></html>
```

3.3. Guidelines and Goals in Project Design

For the projects, anticipated targets are incrementally embedded in the descriptions of the projects so that students can gradually grasp Web technology along the development of the projects.

As what we have discussed previously, the ultimate goal of the class is to create a non-trivial, real world problem solving, three-tire, and web-based application by students themselves. To help students to reach the goal, we not only create labs in assisting them to understand the design issues in the web-based application development, we also provide them with the projects that help them to integrate what they have learned from lectures and abs into their applications. Similar to the design philosophy for labs, an incremental, spiral, topic-driven approach is followed through three projects.

The first project provides students an opportunity to implement a simple but meaningful web-based application. The intention of the project is to let students put basic html components, fundamental Javascript statements, and primary server-side scripting all together to see how a web-basic application can be built. With this overview of the web-based application development, students will gradually realize what they need to learn for the next project as they want to put more advanced functions into their projects.

Examples of the first project could be an online survey, poll, or quiz that uses basic html components, such as *radio button* and *check box*, and with limited server-side process through *form*. As long as the project is meaningful and works, students are often excited and eager to get into the next project.

One of the main guidelines for the second project is to let students use the knowledge they have learned so far and the experience they have gained from the first project to build a non-trivial, three-tire, and web-based application. In addition, the level of the complexity of the second project should be significantly higher than the one of the first project.

the second project requires students to add more functionality on both client and server sides. For examples, on client-side, input pattern checking, online calculation, and standardizing style should be included. On the server-side, database handling must be implemented. Examples of the second project could be an academic conference management system or a textbook adoption system.

Since it adds more requirements than the first project and is a three-tier web-based program for a particular application, the second project shows how an incremental, spiral, and target-driven approach has been utilized in building a web-based application..

The third project is the final project of the class. Since the first two projects are lead by the instructor, the final project is an opportunity for students to apply what they have learned from the class to a real-world problem chosen by them. For instance, students can chose to create a gift shop or video shop website. Although students can still consult with the instructor, all the decisions about their final projects would be made by them. Web components student built for the previous two projects are allowed to be included in the final project.

The guidelines and goals in designing the projects are provided in Table 5 below.

Table 5: Guidelines and Goals in Designing the Projects

1) First project:
- Goal: Designs and implements a two-tier and web-based application with preliminary but enough knowledge in Web technology
- The learning outcomes:
  - Knows the fundamental building blocks of a web-based application
  - Knows the basic components an HTML document has
  - Knows the critical steps in establishing a server side response to a client side request

2) Second learning cycle project:
- Goal: Designs and implements a three-tier and web-based application for a real world problem with the assistance and guidance of the instructor.
- The learning outcomes:
  - Knows how to handle more sophisticated process on the client-side
  - Knows how to use a DBMS to deal with users' data
  - Knows how to carry out a server-side business transaction.

3) Third learning cycle project:
- Goal: Designs and implements a multi-tier web-based application with an intended complexity for a real world problem defined by students themselves.
- The learning outcomes:
  - Knows how to digest the knowledge learned from the class to analyze a real world problem
  - Knows how to integrate the technology obtained from the class to come up with a sound solution to a real world problem
  - Knows how to learn new web technology by students themselves

4. Teaching Philosophy and Considerations

How to determine what topic should be included in the lectures, labs, or projects depends on many factors, such as the instructors' knowledge, current technology, and prerequisites of the class.

In addition to a broad range of topics covered in Web technology, there are many other aspects an instructor should consider before teaching a Web technology course. These aspects are the configuration of web servers, operating systems, and system platforms. System knowledge is as important as language expertise to an instructor. The philosophy we followed in designing and teaching a course in Web technology is summarized in Table 6 below.

Table 6: Teaching Philosophy

Rule 1: cover adequate enough topics It is impossible to cover everything in Web technology in a single course;
Rule 2: know good enough but not everything before teach It is very unlikely for anyone to know everything before to teach Web technology;
Rule 3: know how to find a solution but to know the solution It is critical to know how to find a solution rather than to actually know the solution

Web technology has been changing constantly. It is almost impossible to catch up the changes while you are teaching. Therefore, focusing on fundamental concepts and principles, knowing how to learn based on what have learned, and touching the latest technology have been our guidelines in dealing with the change in Web technology.

Where the class should start with is mainly determined by the prerequisites. To meet what we want, students need to have Java programming experience, database knowledge, software engineering concepts, and operating system or computer network preparation.

5. Conclusions

Following the learning cycle is the fundamental principle we engaged in designing labs and projects. With an incremental, spiral, target-driven approach, students could learn substantial materials in Web technology and be able to build a non-trivial, three-tier, and web-based application in a single semester.

The labs and projects design principles discussed above had been used in teaching a Web technology course from 2000 to 2002 for five semesters and the response was applaudable. Our experience told us that teaching Web technology is very challenge as well as rewarding. Due to its wide range of topics, covering adequate enough topics in a single course will continuously be a challenge to us and it will eventually be decided by the knowledge of instructors and the learning outcomes of curricula.

6.References

[1] W. S. Chung and D. McLane, "Developing and Enhancing a Client/Server Programming for Internet Application Course," proceedings of the Consortium for Computing Sciences in Colleges 2002

[2] J. E. Courte, "The Difficulties of Incorporating Web Development into a University Curriculum," The Dream and Reality WWW @ 10 Conference, Terre Haute, Indiana, Sept. 30 – Oct. 2, 2004 http://www10.cs.rose-hulman.edu/Papers/Courte.pdf,

[3] A. Lee, "A Manageable Web Software Architecture: Searching for Simplicity," proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, Reno, Nevada, February 19 to 22, 2003

[4] J. Liu, "A New Framework for Teaching Web Technology," proceedings of the 2004 International Symposium on Web Services, Las Vegas, Nevada, June 21 – 24, 2004

[5] J. Liu, "Know What to Learn and Learn How to Learn: Towards an Integrated Approach to Teaching Capstone Course," proceedings of The 2005 International Conference on Frontiers in Education: Computer Science and Computer Engineering, Las Vegas, Nevada, June 20 – 23, 2005

[6] J. Phillips and et. al., "Design of a Two-Course Sequence in Web Programming and E-Commerce," proceedings of the Consortium for Computing Sciences in Colleges 2003 (Eastern Conference)

[7] K. Treu, "To Teach the Unteachable Class: An Experimental Course in Web-Based Application Design," proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, Northern Kentucky, February 27 - March 3, 2002

[8] K. Yue and W. Ding, "Design and Evolution of An Undergraduate Course on Web Application Development," the proceedings of ITICSE 2004, Leeds, United Kingdom, June 28-30, 2004