

## USING WEBEX IN A WEB APPLICATION PROGRAMMING COURSE

Natalya Goreva, Indiana University of Pennsylvania, natalya.goreva@iup.edu

Michael V. Yudelson, University of Pittsburgh, mvy3@pitt.edu

Bryan A. Marshall, Georgia College & State University, bryan.marshall@gcsu.edu

---

### ABSTRACT

*Using examples of programming code is a key element in learning how to program. Instructors frequently add comments in the lines of code in order to explain code. This method is effective; however, it has serious complications: first, the code is not “flexible”, i.e. each student has to view all comments at once, even if he or she does not need all of them. This is especially true for large and complicated programming projects. Second, the size of comments, though technically not limited, can be an issue because long comments on multiple lines of code prevent students from having a clear picture of the code. The projects in a intermediate business programming class have both limitations: they include thousands of lines of complicated code, multiple sections and dozens of web pages. The objective of this research project is to apply the method of web-based learning to overcome the limitations of comment-based code learning. The interactive web system WebEx, introduced in one of the business application programming classes, is an example of interactive web-based learning systems that provides example-based code learning, and can be personalized by each student based on her or his individual learning pace.*

**Keywords:** MIS Education, Web-Based Learning, VB.NET, Business Programming, WebEx

### INTRODUCTION: OVERVIEW OF WEB-BASED LEARNING METHODS

One of the most effective methods for student learning is example-based teaching. This is especially true when learning a programming language, where analyzing code snippets is the only available learning tool [2].

There are many successful web-based learning systems that assist in teaching programming languages [2, 3, 8, 1]. These systems are quite diverse in information delivery and feedback methodology (WebEx [2] and QuizPACK [3] are hypermedia systems, while ELM-ART [8] and LISP-Tutor [1] are cognitive tutors). However, they all are designed to

help students understand the programming code. Web-based systems are a self-study tool that students can access out of class, working at their own pace without the need to consult an instructor.

Most often these systems are deployed within Computer Science or Information Systems departments – a “traditional” place to teach computer programming. However, with the growth of the e-Business/e-Commerce industry and the volume of deployed commercial web-services, more and more business majors need professional training in mainstream programming languages [5]. Deploying web-based learning systems that would help these students learn programming thus becomes important.

The goal of this research project is to introduce a method of web-based learning in business application programming courses. Most intermediate and advanced web application programming courses include learning a programming language in the context of a commercial data-driven application similar to real world commercial applications, web projects, and web services. Such web applications include (as a required minimum) a “back-end” component and a “front-end” component with the functionality and content that generally fit the model “Home → Catalog/Product Detail → Shopping Cart → Registration/Login → Checkout” [6].

### WEBEX SYSTEM

In order to place code examples on the system, the instructor prepares code listings and annotations (comments) to the lines of the code. Code examples are served via the system. WebEx shows each code example as an interactive hypermedia learning object (Figure 1). Lines of code have annotations which are clickable. When a student clicks on an annotated line, the instructor’s comment is shown below. WebEx uses a CUMULATE user modeling server [4] to track student’s actions. Lines of code with previously unseen annotations have empty check boxes to the left (Figure 1). Once a student views the annotation, the checkmark appears in the box. When students

return to work with the code examples, all their previous work is shown to them.

After being converted to WebEx, the code which was crowded with comments, transforms into a simplified view where students can now choose which annotations to view (Figure 1). Each line of code that includes an attached annotation is indicated with an empty checkbox. If a student opens the attached annotation, the checkbox is marked with a check, indicating the line was viewed. As mentioned earlier, the information about student views are saved by the system and can be retrieved later by instructor.

### Implementation of WebEx in Course Instructions

In order to properly integrate WebEx into the course, the following guidelines [3] should be followed to motivate students to use the system. The primary goal of the following guidelines is to achieve a consistency of content in the course topics, quizzes/exams, and WebEx:

- (1). The course projects (described in the next section) are divided into smaller modules (e.g., back-end user applications include the modules “Working with Open File Dialog controls”, “Retrieving information from Database”, “Calling SQL stored procedures from VB code”, etc.).
- (2). Each module is demonstrated and explained in class by instructor.

- (3). The code of each module (along with similar examples) is placed in WebEx with annotations for student review (Table 1).
- (4). Each quiz or exam includes questions related to the code discussed in the modules and annotated in WebEx.

The structure of WebEx allows the instructor to use multiple methods to evaluate the frequency of use:

- (1). General monitoring of class activity: when the instructor logs in to the course page, WebEx highlights the frequency of views of each example by changing the color of the letter “D” (Figure 2) next to example name from white (never viewed) to dark green (100% of students viewed the example)
- (2). Detailed individual student activity: by using the system, the instructor is able to measure each student click per page and clicks per time interval.
- (3). Detailed class activity: the instructor can view the same values as in (2) for the entire class.

Previous research shows that implementing similar interactive hypermedia tools in programming courses has increased student performance of the programming languages by 50% [3].

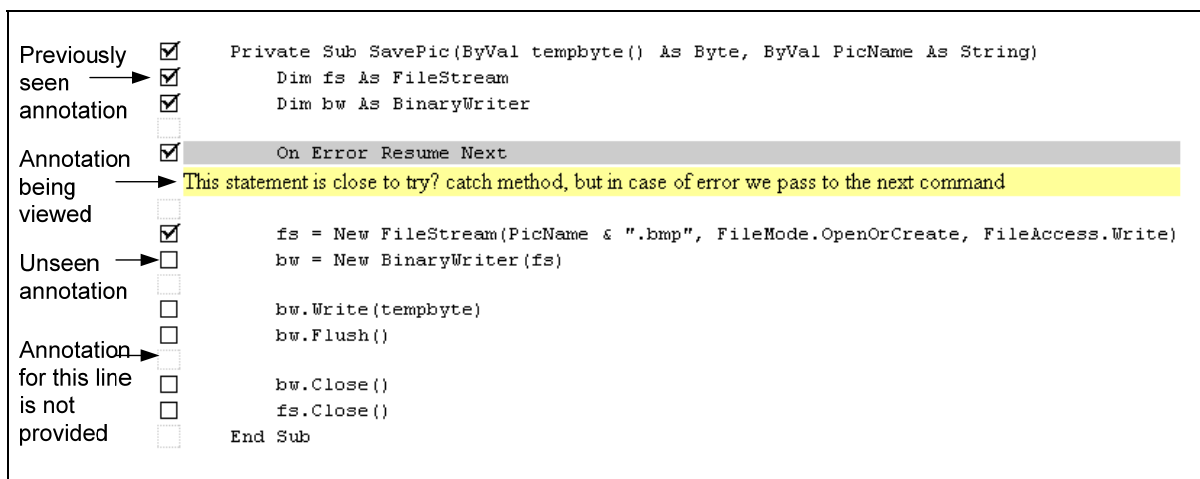


Figure 1. Annotation of Lines Code Example in WebEx

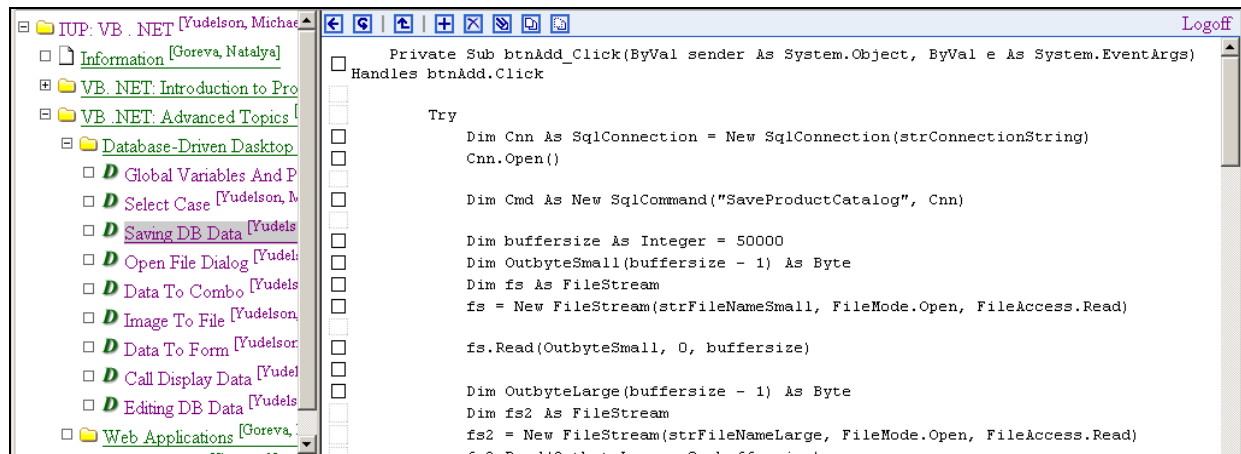


Figure 2. WebEx: Instructor’s View

## PROGRAMMING CONTENT

The system was implemented in an intermediate VB.NET web application programming course at Indiana University of Pennsylvania. Apart from a general VB review session, this course included four stages: desktop application, project database, web application, and web service development (Figure 3).

The purpose of the course is to teach students to develop all of the components for a web application written in VB.NET. The final projects that students demonstrated at the end of the course emulated commercial web sites designed to market products online. Each project included a database with the product data (Product Catalog) and the customer data (Customers Personal Data, Past/Current Orders, Session Shopping Cart Info, etc.). The databases were located on the course server, and were designed to exchange data with both the desktop and the web applications (Figure 3).

In the project model shown in Figure 3, the desktop application was needed so the back-end user (authorized company employees) can view company inventory and other data, and also be able to make changes to the product catalog in particular: adding, editing, and removing products.

The web application was an actual e-commerce site where users could search for products, add products to their cart, login/register, and purchase products. The structure of the experimental web site was designed based on the model in prior research [6]. Students were working on their individual projects,

which included (but were not limited to) the same functionality.

Finally, the web service was used by the web application to verify the user’s credit card number. The credit card verification in the course project follows the same pattern as the real verification process: when a user enters the credit card information, it goes to another server (in real world case, the server of a credit verification web service provider), where her or his credit card is evaluated using a verification algorithm (in our case, only the Luhn algorithm [7]) and charged. The result of a verification is returned to the main server as a value *true* (accepted) or *false* (declined) as seen in Figure 3.

## Research Hypotheses

RH1: A higher frequency of WebEx use will result in higher student performance in the web application programming course.

RH2: A higher amount of WebEx course coverage viewed will result in a higher student performance in the web application programming course.

The independent variables are (1) frequency of use (measured in clicks) and (2) course coverage. The dependent variable is the improvement student performance on tests.

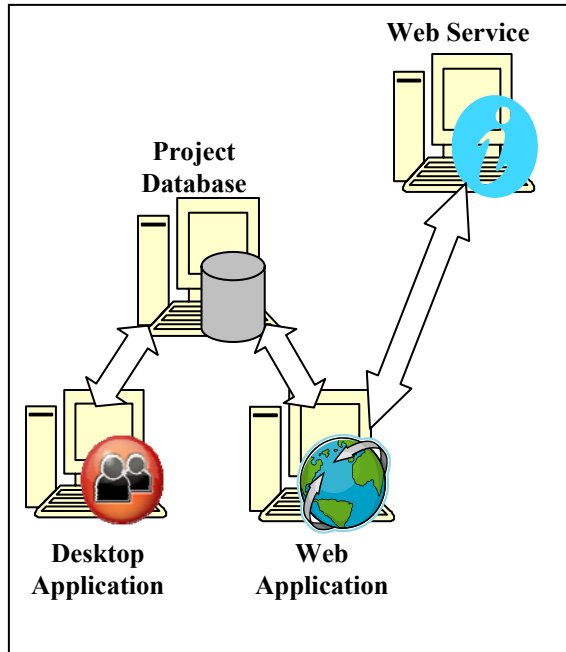


Figure 3. Connection Among Project Components

## METHODS AND PROCEDURES

### Measuring Frequency of Use

Frequency of WebEx content usage was measured by counting the number of clicks made by each student on course examples captured by the CUMULATE server.

### Measuring Course Coverage

Course coverage was measured as the number of course topics viewed by the student. This value is not the same as the frequency of use. For example, if a student has difficulty understanding data connections and returns to this section multiple times, the frequency of use will be high because of the time spent on the system. However, the coverage value will be low because of the low number of topics viewed. It is important to see if each of the two factors (frequency and coverage) impact student performance.

### Measuring Student Performance

The student performance variable was based on the relative value of the students test scores. It was calculated with the following formula:

$$\Delta_r (\%) = 100 \% * ((\text{Test2} - \text{Test1}) / (100\% - \text{Test1}))$$

If there was not improvement (second test result was the same or worse than the first),  $\Delta_r = 0$ .

For example, if a student earned a 76% on the first test, this means that he or she did not understand 24% of the course material. If the same student's score on the second test was 84%, we assume that out of the 24% of the course content that was not understood, the student was able to catch up on 8% of it, and the relative improvement in this case is:

$$33\% = 100\% * ((84\% - 76\%) / (100\% - 76\%))$$

### Procedures

The class content was divided into two equal parts and the subjects understanding of the code was measured at the end of each part, via a test. WebEx was introduced in the second part of the course and students were encouraged to use it as a source for code examples and explanations.

The rest of the course material and the teaching methods remained the same as the first section. Both the first and the second test were based on code explained during the course and were designed to check students' understanding of this code.

Frequency of use and course coverage were computed for each student. The value of student performance was calculated based on the procedure described in the previous section.

A one-sample *t*-test was used to compare the means of the scores on the 1<sup>st</sup> and the 2<sup>nd</sup> tests. In addition, a correlation test was run before testing RH1 and RH2, in order to check if *frequency* and *course coverage* correlate with *student performance*. Finally, regression tests were applied to test RH1 and RH2 (higher frequency of use and higher course coverage of interactive code examples results in higher student performance in a web application programming course).

## RESULTS

### Difference of Means

The average test score on the second test was 13.5 points higher than the average score on the first test, when WebEx was not implemented (74% vs. 87.5%). The *t*-test run on the test results indicated that there was a significant (0.01 level) difference between the scores. Shown in Figure 4 are the distributions of the scores on test 1 and test 2 based on the score ranges < 60% (F) to >= 90% (A).

**Correlation Tests for RH1 and RH2**

A correlation test was run on Frequency/Coverage and Student Performance which showed the following results:

- (1). There is a strong correlation (0.82, significant at the 0.01 level) between the number of clicks on WebEx examples and the increase in student performance.
- (2). The coverage on WebEx and the student performance tend to correlate, but without statistical significance (0.62 with sig = 0.053).

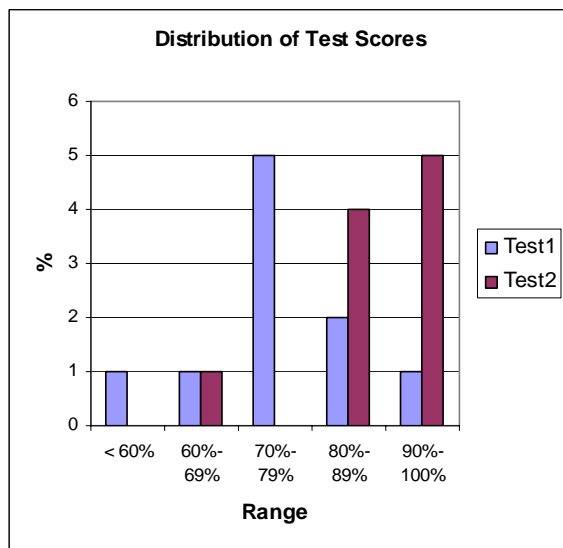


Figure 4. Distribution of the Test Scores by Ranges

**Regression Analysis for RH1 and RH2**

Four regression tests were completed to see what part of the variance in subjects' performance was explained by the frequency of use and the amount of coverage viewed on WebEx. It was found that the frequency of use of WebEx examples explained 68% of the variance in student performance. The amount of course coverage explained 57% of the variance in student performance. All these results were found to be significant at the 0.05 level. Hypotheses RH1 and RH2 were not rejected.

**Student Performance by Groups**

Though not included in the research hypotheses, it was interesting to see how the student performance was impacted by frequency of use and course coverage, grouped into low-, medium- or high-

groups, respectively. The following tendencies were discovered (Figure 5):

- (1) For the low-frequency and low-coverage groups, the values of both absolute and relative improvement were the lowest.
- (2) For the high-frequency and high-coverage groups, the values of student performance were the highest.

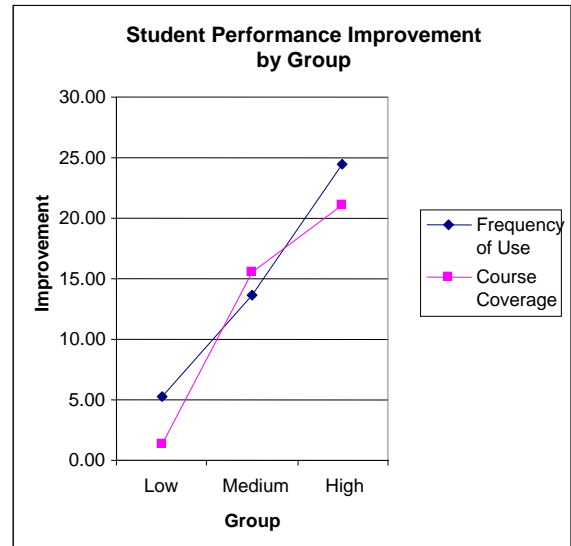


Figure 5. Student Performance Improvements by Groups

The statistical tests were not run on this data because of insufficient numbers of subjects in each group; however, the tendency found could guide future research.

**LIMITATIONS AND DISCUSSION**

There are several limitations to this research. First, due to an insufficient number of students it was not possible to run analysis of variance by the groups of subjects with low, medium, and high frequency of use and amount of content. With the intention to implement the WebEx systems in more business programming courses, such tests will be possible in the future.

It has been suggested that the greatest effect may be reached if WebEx is used throughout the entire course [3]. Therefore, in future courses, WebEx will include more examples, all-course coverage, and more content in general.

## REFERENCES

1. Anderson, J. R. & Reiser, B. J. (April 1985). The LISP tutor: it approaches the effectiveness of a human tutor. *BYTE* 10(4), 159-175.
2. Brusilovsky, P. (2001). WebEx: Learning from examples in a programming course. In: Fowler, W. & Hasebrook, J. (Eds.). Proceedings of WebNet 2001, World Conference of the WWW and Internet, Orlando, FL, AACE, 124-129.
3. Brusilovsky, P. & Sosnovsky, S. (2005). Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK. *ACM Journal on Educational Resources in Computing* 5 (3), Article No. 6.
4. Brusilovsky, P., Sosnovsky, S. A., & Shcherbinina, O. (2005). User modeling in a distributed e-learning architecture. Paper presented at the 10th International Conference on User Modeling (UM 2005), Edinburgh, Scotland, UK, July 24-29, 2005.
5. Bureau of Labor Statistics, *Fastest Growing Occupations, 2004 – 2014*, Retrieved 7/13/07 from <http://www.bls.gov/emp/emptab21.htm>
6. Goreva, N. (2006). Product information factor in online buyer behavior when shopping for high-cost and low-cost products (doctoral dissertation). Utah State University, Logan, UT.
7. Kilgo, J. (2003). *Using the Luhn algorithm*. Retrieved 3/10/07, from <http://www.dotnetjohn.com/articles.aspx?articleid=97>.
8. Weber, G. & Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education* 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, 351-384.

NOTE: A special thanks to the University of Pittsburgh for the use of resources relating to this study and the WebEx software. The portal to the software can be found at the following link:

[http://kt1.exp.sis.pitt.edu:8080/portal\\_client/](http://kt1.exp.sis.pitt.edu:8080/portal_client/)