Teaching Web Services and Service-Oriented Architecture using Mobile Platforms

Mark Sherriff
University of Virginia, sherriff@virginia.edu

Abstract – **Even as everyday software increasingly transitions to online software services, many educational projects in software development courses still remain on single-user PC systems. To better prepare our students for this emerging trend in software development, material on web services and service-oriented architecture should become more of a focus in computer science courses. This paper describes our experiences in teaching a senior-level course at the University of Virginia in the fall of 2009 geared around the concepts of web services and service-oriented architecture using the Windows Mobile platform. In this course, students worked in teams to produce web services that could be utilized by other teams to eventually build a campus tour application for the Windows Mobile device. We believe that using mobile devices is an optimal vehicle for teaching service-oriented architecture due to their relatively limited processing power, portability, pervasiveness in society, and interest among students.**

Index Terms – Service-Oriented Architecture, Web Services, Mobile Devices

INTRODUCTION

Emerging software systems are trending toward integration into and dependence on information and functionality being available online constantly [1]. Even software systems that were traditionally considered single-user, single-computer systems, such as word processing or spreadsheet management, have become quite popular as web applications, such as through the Google Docs application website [2]. While there is certainly still a large market for the stand-alone versions of these applications, the next generation of computer users is viewing these web applications as part of the norm as more and more applications become available.

Web services are defined as a "self-describing, self-contained software modules available via a network, which complete tasks, solve problems, or conduct transactions on behalf of a user or application" [1]. While web services can appear in different forms and in many different programming languages, they, in general, represent a business- or feature-oriented task, encapsulating all the code for a single feature or business process. Web services, as a part of a service-oriented architecture, encourage developers to build their systems in a more modular fashion, leveraging the abstraction of the services to make code and feature reuse more viable.

Popular web applications have started to make their web services available for others to use for various purposes. Google Maps is a prime example, as the API for the mapping application has been used to create all sorts of interesting and useful mash-up applications [3]. Flickr's web service API provides some basic image editing features, such as red-eye reduction [4]. These growing numbers of web services are providing opportunities for collaboration among developers to create the next suite of applications.

Meanwhile, mobile technologies are growing exponentially around the world. Not only is the pervasiveness of smart-phones growing in general, mobile computing is growing in places that previously had little to no computing capability. In areas of the world where Wi-Fi is unheard of, a cellular signal might still be reachable. Medical personnel have been using mobile technology and text messaging to do remote diagnosis of patients in hard-to-reach areas of Africa [5]. Stories like this inspire students to build the next generation of mobile computing application. This is an area they want to be exploring in their education.

Amidst this backdrop of expanding advances in web technology and services, we still see a lack of education regarding these technologies in our computer science programs as departments try to keep up with the trends [6]. Students come into the computer science major eager to work with these technologies that they themselves have used and may have had some limited experience working with. Unfortunately, when departments haven't caught up to the current trends, students become disappointed and might eventually leave the major.

We face a similar scenario at the University of Virginia in the Computer Science Department. We saw several holes in our curriculum in these emerging and growing technologies and aimed to fill them first with special topics courses, which could later be permanently added to the curriculum. This paper describes a senior-level course taught at UVa in the fall of 2009 geared around the concepts of web services and service-oriented architecture using the Windows Mobile platform. In this paper, we will describe the course, it's project and assignments, and how effective it was in preparing students for building these new service-oriented, mobile software systems.

Sections 2-4 of this paper illustrate the background of the course and the project itself. Sections 5 and 6 present our lessons learned and how to incorporate our methods into

978-1-4244-6262-9/10/$26.00 ©2010 IEEE

other courses. The remaining sections describe the reception of the course by the students, including continuing research projects, and our conclusions.

COURSE BACKGROUND

I. Course Organization

Our course is called Web Information Systems Engineering and is currently being taught as a special topics course in anticipation of being fully integrated into the curriculum after review. The course was first offered in fall of 2009 and had 54 students total. The course will be offered again in the fall of 2010. Students meet twice a week for 75 minutes for lecture and there is no lab component to the course. We required that students had completed our data structures course as a prerequisite. We did not require advanced software engineering since we cover the basics of the development lifecycle early on in the core curriculum. Thus, a number of students in the course had only experienced a minimum of rigorous software engineering.

We as a department feel strongly about providing students the skills required to work effectively in teams and have shown some of the advantages of teamwork in courses [7-8]. Teamwork is a specified focus in our software engineering and other software development courses and that philosophy carried through to this course. Students were allowed to name up to three students they would feel comfortable working with and up to three students that they felt they could not work with successfully. The course staff then assigned the groups, taking their preferences into consideration along with previous performance in courses.

II. Course Technology

For this course, Microsoft Academic Relations provided us twenty-five Windows Mobile devices. The set of phones was comprised of Samsung Epix phones and HTC Fuze phones. These two devices both had similar capabilities: touch screen with stylus, Wi-Fi, GPS, and Windows Mobile 6.1. Windows Mobile turned out to be a good choice for the project as C# was relatively easy for our students to learn and the toolset and GUI builders were intuitive.

For the web services aspect of the course, we used a single Ubuntu server running Apache's Axis2 software. Students were also allowed to create their own POX (Plain Old XML) services in PHP if they so desired. We also took advantage of Amazon Web Services through their academic program.

III. Course Topics

We endeavored to cover a wide range of topics applicable to web services and mobile development. The topic list included:

- Client Software Architecture (Firefox, Chrome, IE)
- Client Hardware (Desktop vs. Mobile)
- Anatomy of a Web Application
- PHP / MySQL
- LAMP Server Setup and Maintenance
- RESTful and Non-RESTful apps
- Web Apps vs. Client Apps
- Embedding Apps / Mash-ups
- Service-Oriented Architecture (POX, SOAP, etc)
- Enterprise Architectures (J2EE, .NET)
- Location-Based Services
- Peer-to-Peer Architecture
- Security and Trust Management
- Privacy
- Agent-Based Systems
- Usability and Accessibility

COURSE WEB SERVICES AND MOBILE PROJECT

The course is organized around a semester-long project in which teams of three students build a number of software services and a final mobile application that utilizes these services. Our goal was to create a large enough project that could be broken up into sub-lessons and sub-assignments during the course of the semester. Anecdotal and evaluation feedback from students in prior courses indicated a reluctance to continue to "start from scratch" at the beginning of each assignment. Further, we wanted the project to be something that took advantage of the mobile devices' capabilities while also was interesting and engaging to the students.

During discussions with the students early on in the course, there was a desire to create an application that would be useful to other students at the University of Virginia. The students were also interested in using the GPS capabilities of the phones for some purpose.

We decided as a class to build a tour application for students at or visiting UVa. The main features of the application can be found in Table I below. The project was split between core and optional requirements. Teams were required to complete all core requirements and at least three optional requirements. To provide teams with greater freedom of creativity, teams were allowed to propose their own optional requirements as well. These requirements simply had to be authorized by the course staff to ensure they were of similar difficulty to the other requirements.

Many of these requirements were based around various web services. After the students had performed a basic homework assignment in learning C# and the API for using the GPS on the phones, the students were tasked with building their first web service. The initial web service would take a set of GPS coordinates as a parameter and would return a listing of the five closest buildings. To populate our class database with building information, each team in the class was given a list of three or four buildings around campus. Teams then entered their building GPS coordinates, history, picture, and other information into a single database shared by the class. Students created their first web service using either Axis2 or POX and hosted it on our class server.

Like many departments, our introductory courses in programming are in Java, and thus the Axis2 environment

proved to be a good choice. Students were quickly able to grasp the use of the available tools and could create working web services after a few attempts. Some advanced students who had more web programming experience wanted to create their services using PHP and XML. After covering the differences between the technologies in class, many students continued on with these technologies and also build working web services.

TABLE I
REQUIREMENTS FOR COURSE PROJECT

Requirement	Core or Optional Requirement
The system shall have a startup screen displaying the authors of the application.	Core
The system shall display a main screen after the startup screen to provide users with a list of the main options and features in the application.	Core
The system shall use the GPS capability of the mobile device to list the five closest buildings to the device's current position.	Core
When a building is selected, the system shall display a picture of the building along with information about its history, what departments are located there, and any other pertinent information.	Core
The system shall update the current GPS coordinates of the device either automatically or by the request of the user.	Core
The system shall allow a user to search through the database of buildings and to view any building in this manner.	Optional
The system shall allow users to input a pre-determined list of buildings in a specified order to create a tour through campus.	Optional
The system shall show the user's current position, along with the position of the closest five buildings, using either Google Maps or Bing Maps.	Optional
The system shall allow users to search for and find other campus amenities, such as vending, dining, libraries, and dormitories.	Optional
The system shall show the location of the closest bus stops, along with the time schedule for the next bus to arrive at that location.	Optional

After their first web service was created, it was integrated into their initial version of the mobile software. Students then moved on to the next web service, whether that was map integration or searching for campus amenities or something else, and that web service was incorporated into the mobile software. Students were encouraged to share data between teams, such as GPS coordinates of locations on campus and other services, to create better applications. Each week, students were required to give the course staff a brief demo of their project, showing what had been completed. Thus, there was a distinct rhythm in the course of creating a web service and then integrating it into the mobile platform and then repeating until the project was complete. The end result was a software system consisting of a rich client application on the mobile device coupled with XML-based web services on a shared class server.

At the end of the class, the students were given an oral final exam. Each student received four randomly chosen questions, from which they had to answer three. Below are some example questions, portraying the level of knowledge we expected students to have at the conclusion of the semester:

1. Fully compare and contrast SOAP and POX as web service technologies.
2. Explain REST and be able to give and defend RESTful and non-RESTful web applications.
3. The original model of a web services architecture was the Publish / Find / Bind model. Describe this concept and explain how it was/could be implemented.
4. Describe the three-tier architecture and how it is used in both web clients and stand-alone applications.
5. Compare and contrast thick vs. thin vs. rich internet application for mobile devices. Why would you use one over another?
6. Discuss the concepts of authorization, authentication, integrity, and non-repudiation with regards to web services. Be able to describe ways of implementing each.
7. Discuss application signing with mobile devices. Talk about how this adds a layer of security to mobile applications.
8. Discuss some usability challenges with mobile development.
9. How does a SOAP envelope work? Why use SOAP at all? What's the benefit?

EXAMPLE PROJECT

Below is an example project created by one of the teams in the course. This team performed well in the class and created a well-designed application that is still in use by some in the school. Figures 1 and 2 show some screenshots of the application.

FIGURE 1
MAIN SCREEN OF EXAMPLE PROJECT.

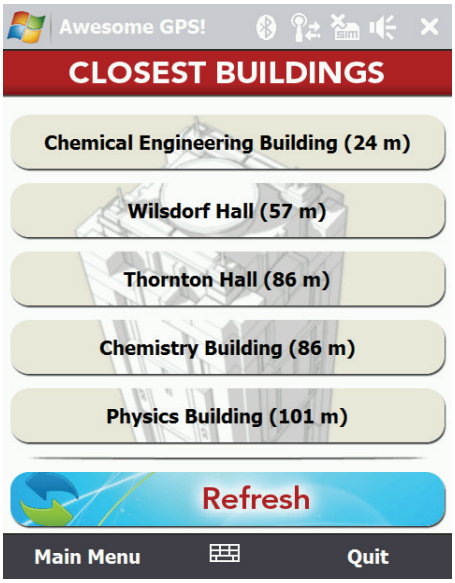

FIGURE 2
CLOSEST BUILDINGS SCREEN.

The example project shown here also incorporated Google Maps, searching capabilities, and the ability to find campus dining establishments based on user criteria.

TEACHING WEB SERVICES

Other departments have used innovative methods for teaching web services to their students. Nandigam et al. describe their tool called WSExplorer [9]. This tool is geared to connect to various web services and to dynamically show students exactly what is occurring when the connection is taking place. Due to the nature of web services often being closed, black-box-type systems, there is great value in illuminating how a SOAP transaction takes place for students. Students are already accustomed to debugging local code by printing out information as it appears, and this approach makes it significantly easier for students to work with web services. Nandigam et al. report that students enjoyed using the tool, as it made the lessons more engaging [9]. Our goal is similar in that we want to increase student enthusiasm and interest in web services. Without some mechanism to reveal what is occurring when a service is called, it is possible that students will dismiss the technology. We feel that using mobile devices also help to educate students as to how services operate, as they have to deal with transmitting data to a lower-powered device.

Many departments differ on what language to use to teach web services. Kachru and Gehringer examined the positives and negatives of teaching web services using .NET versus J2EE [10]. Their efforts revealed that both platforms are viable options, as .NET and C# are still on the rise, while Java in general still has a prominent place in industry. Both environments have good tools and resources, as well as academic support from their corporate entities. In our course, we used a combination of the two languages. All development on the mobile devices was done in C# for .NET. Since the phones all ran Windows Mobile 6.1, this was a natural choice. Many students implemented their web services (and some smaller test clients) using Java in Apache's Axis2 environment. Further, lectures were presented comparing and contrasting the two technologies so that students would be aware of the differences in the future.

Kendall and Gehringer address the problem that web services themselves are quite complex and hard for students to grasp in a short amount of time. In a web topics survey course, web services can often only receive a small portion of the time. To address this deficiency, Kendall and Gehringer created the Water programming language for web services [11]. The Water language is designed to minimize the learning curve of web service technologies by compartmentalizing many of the standard functionalities in an easy-to-use XML based language. While the students found the language approachable, there was some concern among the students as to how useful their skills in the language would be going forward into industry [11]. We certainly agree with Kendall and Gehringer that the technologies behind teaching web services can be quite complex, particularly if the topics have to be covered in a short amount of time. Since our course was for a full semester on web services, and not a section of a larger web topics course, we were able to go into more depth into each of the various technologies to help make them clearer to students.

While courses on web services and service-oriented architecture are certainly evident in curricula in computer science, we believe that our methodology of integrating mobile technology is novel to this type of course and adds a great deal to the course as a whole. Using mobile devices is an optimal vehicle for teaching web services and cloud computing concepts because of the pervasiveness of the devices in society, the broad range of possibilities, and the good platform support in .NET for web services. All these factors combined help create a learning environment in which the students are excited to be using current technology that is improving skills that they can use immediately, while also not being too daunting for them to achieve in a single semester. Further, with the rise of the mobile platform as a major environment for new and innovative software, students are becoming increasingly motivated to learn how to program for these devices.

In our approach, we tried to focus on the web services aspect of the course as much as the mobile aspect, to ensure that this did not become solely a mobile development class. The cyclical nature of the project, creating a web service and then implementing it into the mobile application, seemed to facilitate that as best as possible.

LESSONS LEARNED

One thing that was confirmed in our minds through this course is the attractiveness of flexibility in project-based courses. Students want to have input in the software that

they are creating and often their ideas are better than those of the course staff. While it can make grading more difficult, if the optional paths that students can take during the course of the project are kept as equivalent as possible with regards to difficulty, the students seem to get that much more excited about the course and the project.

Another positive aspect of this type of project is the idea that students were building something that they themselves or their friends could use. The idea of a service-learning software project is certainly not new [12], but it bears repeating that students care much more about projects that have meaning behind them, as opposed to contrived examples.

BUILDING YOUR OWN COURSE

We encourage other faculty members to merge mobile technologies with their web service courses. To this end, we will provide some guidance in building these courses in this section.

Probably the first issue to address is the availability of mobile devices for education. While this may seem to be a difficult task, there are some avenues to explore. Many companies, Microsoft and Google included, have specifically started programs to get their devices into the hands of students for educational purposes. While we used Windows Mobile devices, Android devices would also work quite well, particularly for programs that start with Java as their first programming language. The Android development environment is available for free from Google and is incorporated directly into the Eclipse integrated development environment.

Another platform to consider is the one found on iPhone and iPod Touch devices. These devices are of increasing popularity among students, and it's likely that many students already have one of these two devices. Apple provides iPhone and iPod Touch development support for educational purposes and makes it possible for students to install their own applications on their personal devices. The drawback here is that the Apple development environment, XCode, must be run on an Apple-branded computer. If your department or school does not have a Mac lab of some kind, this might be hard to accomplish.

Even if actual phones cannot be obtained, the development environments and emulators for Windows Mobile and Android phones are robust and can allow for team development without the access to the device. Faculty could acquire a small set of phones for checkout purposes as well.

We recommend integrating the mobile devices as early as possible into the course. One thing that we found at the beginning of the course was that in starting with the basics of web technology first without the phones, the students got somewhat impatient with having the devices and not getting to use them. While this was not a huge problem overall, it is something to be considered.

COURSE RECEPTION

Students were surveyed at the end of the course to gauge their reaction to the course and the presented material. In general, the course was considered a success by the students, with an average score of 4.52 out of 5 when asked whether this course was a worthwhile course, with 5 corresponding with "Strongly Agree." Further general comments about the course can be found in Table II with further evaluation statistics in Table III.

Like any course taught for the first time, however, there are certainly aspects that need to be improved in the next iteration of the course. Due to the nature of the project, we needed to use a fair amount of database technologies. While some of the students had had some database experience, many did not. We added a database "crash course" section to the topics list, which was not quite sufficient enough to bring all the students up to speed on the skills they needed. Many picked up what they needed on their own, but this aspect of the course needs to be reconsidered. In the future, we will at the very least ensure that there is one students experienced with databases on each team.

Another topic that did not work well was on agent-based systems. The goal was to show students how some web services could run effectively automatically to achieve their goals, but toward the end of the semester, students were focused on completing their projects and this topic did not meet our expectations.

Overall though, many students expressed their satisfaction with the coverage of web services, service-oriented architecture, and enterprise architecture, and have reported already used their knowledge in job and internship interviews.

TABLE II
GENERAL STUDENT COMMENTS REGARDING THE COURSE.

"This should be a required course, its quite awesome and really helpful when applying for jobs."
"The course was the most worthwhile course I have taken in the CS department. A lot of questions during technical interviews pertains to the topics covered in this course."
"This class was exactly what I was hoping for. It fills a gap in my education in a topic that is both highly relevant to the real world and something that I probably wouldn't have taken the time to figure out anyway since I don't have the resources or didn't know I had the resources. Who knew I had access to a mysql server through the university?"
"As one of the only courses I've taken in the cs department that teaches in depth a language other than java or c++, this course helped my resume and job search. The course is also one of the few courses that results in a finished product we can put in our professional portfolios and show to employers. Very useful course, would recommend to friends."
"A wide variety of material was covered intelligently and at a good pace to provide a sample of everything to us. The phone project is a great concept and helped by providing a chance to go through the development process in a team and make design decisions for a tangible application."
"Far too rarely are professors at UVa teaching the newest trends in industry. It was very nice to have a course about what's really happening in the real world as we all prepare to go out and get jobs. I also liked that the students were able to generally guide some of the lectures. Picking several topics that were interesting to us personally was very nice."

TABLE III
COURSE EVALUATION SURVEY RESULTS.

Question	Value (5: Strongly agree; 1: Strongly disagree)
How accurate is this statement for you: The project was of acceptable length.	4.48
How accurate is this statement for you: The project was of acceptable difficulty.	4.22
There was a reasonable level of effort expected for the credit hours	4.21
The subject matter was challenging.	3.78
Overall, this was a worthwhile course	4.52

EXTENDED PROJECT WORK

Two extended research projects have come out of projects from this course. One project involves creating a recommendation system for prospective student tours at UVa. As a student visits various locations around campus using the application on their mobile device, their tour is recorded and presented back to them so they can save it for later. If they grant permission, that data is also added to a database of tours along with some non-identifiable information regarding what they were most interested in their trip to UVa. Then, if other prospective students use the application and state what they are interested in, the application can recommend a specific tour for the student.

Another project is using a web service for using social networks to find lost pets. A service is connected to the camera application on the device, which allows for the geo-tagging of an image and allows for others to use a search service to hopefully find a pet that has been found and photographed by another.

CONCLUSIONS

In this paper, we have presented a course on web services and service-oriented architecture that utilized mobile devices as the vehicle for teaching that material. We found that while there are challenges in obtaining the devices and managing the project, the course is worthwhile and well received by the students. We believe that teaching web services using mobile technology is a natural fit, providing a platform that encourages student participation and excitement. With excellent toolsets available from major mobile device providers, even programs that cannot obtain a full class set of devices can incorporate some of these techniques through device emulation. In the future, we plan to introduce these techniques in earlier courses in our curriculum to try to engage students in solving these real-world challenges with emerging technologies.

ACKNOWLEDGMENT

Many thanks to Microsoft Academic Relations for providing the phones for this course. We would also like to sincerely thank the students of the first section of Web Information System Engineering at UVa for their comments and constructive criticism of this course. We would like to particularly thank Dan Tran, Akshay Joshi, and Steven Mond for providing screenshots of their application for this paper.

REFERENCES

Papazoglou, Michael P., *Web Services: Principles and Technology,* Pearson Education, 2008.

Google, *Google Docs,* http://docs.google.com/, 2010.

Google, *Google Maps API,* http://code.google.com/apis/maps/, 2010.

Yahoo, *Flickr Services*, http://www.flickr.com/services/api/, 2010.

Greengard, S., "Upwardly Mobile," *Communications of the ACM,* Vol. 51, No. 12, pp. 17-19.

Yang, C.-S, "A two-tier on-line learning model for teaching Web services," *3rd International Conference on Information Technology: Research and Education, 2005,* pp. 40-44.

Horton, T.. and Knight, J., "Evaluating A Software Engineering Project Course Model based on Studio Presentations," *ASEE/IEEE Frontiers of Education Conference 2003,* pp. 21-26.

Sennett, J. and Sherriff, M. "Compatibility of Partnered Students in Computer Science Education," *The 41st ACM Technical Symposium on Computer Science Education 2010,* pp. 244-248.

Nandigam, J., Budivada, V.N., and El-Said, M., "Teaching Web services using WSExplorer, " *ASEE/IEEE Frontiers of Education Conference 2007,* pp. S3H-20 – S3H-25.

Kachru, S. and Gehringer, E.F., "A comparison of J2EE and .NET as platforms for teaching Web services," *ASEE/ISEE Frontiers of Education Conference 2004,* pp. S3B-12-17.

Kendall, M.D. and Gehringer, E.F., "Teaching Web Services with Water," *ASEE/IEEE Frontiers of Education Conference 2006,* pp. 7-12.

Layman, L., Williams, L., and Slaten, K., "Note to self: make assignments meaningful," *ACM SIGCSE Bulletin,* vol. 39, no. 1, pp. 459-463.