

**Return address removed to
protect one person's privacy.**

Date and Addressee information removed to protect one person's privacy.

Dear Sir or Madam,

I am writing to offer an insight into my experience in working under Dr. Russell Robbins, ...**Information deleted to protect two individuals' privacy...**

to implement changes in two Web-based applications used in a classroom setting by students to evaluate ethical situations.

Configuration

The applications are JEE projects that operate under Tomcat 6.x with a MySQL 5.1 database provider. The applications are stored on the server via a SubVersion (SVN) repository. This combination of software lends itself well to using Eclipse as the Integrated Development Environment. To do the development work, I installed versions of MySQL and Tomcat 6.x for the Windows 32bit operating system. Eclipse required additional plugins (Subclipse, Maven) to interact with the Subversion version control system and to simplify the management of libraries and deployment. I also installed Apache v2.2 and PHP in order to install phpMyAdmin to administer the MySQL server. I replicated the tables from the server and had a functional equivalent local development environment similar to the production server.

Application Description

The two applications, named *SIMULATE* and *SIMULATE:maven*, are similar in functionality, but use slightly different frameworks. Both applications have a user authentication front-end, and allow a user (or users) to choose from a list of 'cases' to participate in. As illustrated in Figure 1, the user is then guided through a series of roles and situations and has to choose from several decisions associated with them. The selection then influences the next role/situation until the simulation ends. After each decision, the application prompts the user for information regarding the “values” used in making them.

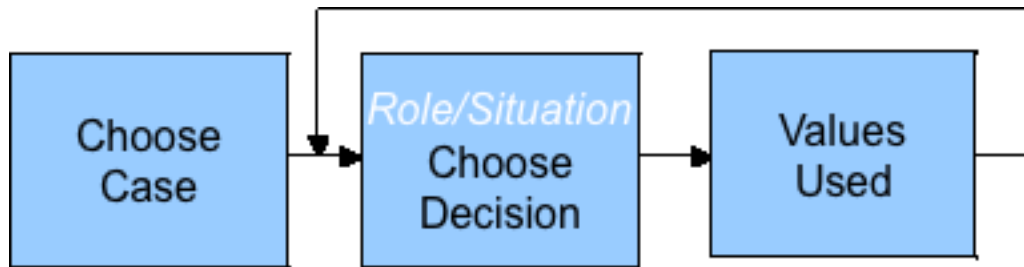


Figure 1: Flow Diagram

SIMULATE:maven is the simpler of the two applications: it is a single-user version where the user plays through all roles/situations in each case. It is implemented using a straight-forward JSP/Servlet architecture, with a POST method on each page sending its data to the corresponding Servlet to be processed, and then redirected to the next JSP page (see Figure 2). The JSP pages are responsible for formatting the page, and the Servlets serve to parse the returned data and update the database. All cases/situations/roles are dynamically served from the database, as opposed to hardcoded in the JSPs.



Figure 2: *SIMULATE:maven* framework

SIMULATE, however, uses a slightly different method in order to update the webpages. *SIMULATE* has multi-user, multi-terminal capabilities and uses an AJAX-like Javascript framework in order to keep separate browsers synchronized. It accomplishes this by rewriting an inner DIV on a simpler HTML page as needed. This allows the server to receive changes from one client and push out updates to multiple other clients (See Figure 3.) Again, all cases/situations/roles are dynamically served from the database, as opposed to hardcoded in the JSPs.

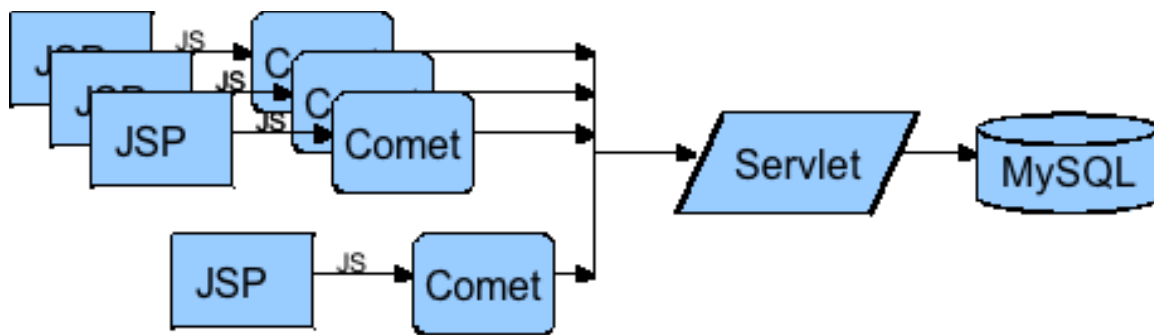


Figure 3: *SIMULATE* framework

Development

The changes requested involved altering the flow (see Figure 1) to include additional elements for data collection. Both applications required modifications to the JSP files and Servlets, as well as the creation of additional tables in the database, to handle the new information. Alterations to *SIMULATE:maven* were relatively straightforward: JSP pages were created or altered to accommodate the new fields. The Servlets used a common class to handle database updates, and the majority of changes happened within there.

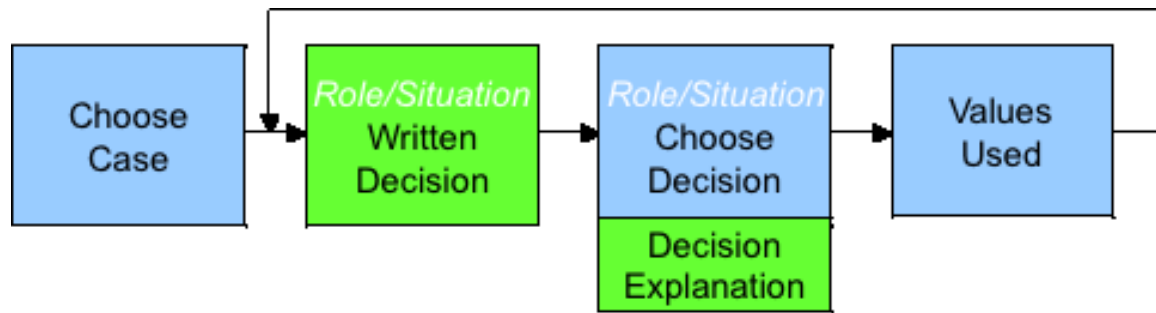


Figure 4: Changes to flow

The modifications to *SIMULATE* were a bit more extensive. The javascript libraries to support the DIV rewriting were generic enough to use mostly without modifications. The underlying Servlet logic, however, required considerably more planning and alteration. Care had to be taken during implementation to maintain synchronization between clients. As in *SIMULATE:maven*, all database operations are handled in a separate class file, and as such, similar functionality had to be added.

Additionally, changes to both applications were implemented to allow easier transitions between development and production environments. Database names and credentials, that had previously been hardcoded, were transferred as environmental variables with the Tomcat configuration files. Other alterations were made that were either low-impact or to improve readability of the code.

Testing

Once primary development had ceased, compatibility testing under several web browser and operating system environments was the next goal. Fortunately, having access to both a Windows and Macintosh computer made this considerably easier. A test matrix was created against the most popular browsers on each platform (Safari, Chrome, Firefox and Internet Explorer) and compatibility was investigated and noted. Most problems involved minor CSS rendering differences between browsers.

Future Development

Dr. Robbins and I discussed several additional changes will need to be incorporated into the applications. An issue that arose in the development of the above addition to the *SIMULATE* application is how the current system breaks under normal user expectations. This is an unfortunate consequence of a static page with javascript DIV rewriting – use of the browser navigation tools (back-button, bookmarks) do not take into account dynamic page changes and delivers unexpected results. In the *SIMULATE* application, for example, pressing the browser's back-button while into the 2nd situation of a multi-part case puts the user at the case selection screen instead of the previous situation as expected. In situations like this the normal approach is to add scripted navigational elements to the page itself and display a warning when the user attempts to use the built-in browser tools. After discussions between Dr. Robbins, **[Name Removed]** and I we decided this would be the best course of action.

Casebuilder

An additional facet that was included into the applications is the ability for user-generated cases. This would work in concert with the navigational system and would allow the user to easily edit all the necessary parts of a multi-situation, multi-role case. Since the amount of permutations would grow at an exponential rate, a robust system needed to be in place to successfully add additional cases. Following extensive discussions and simulated user test cases on how best to implement such a system, our team finally settled on a separate application we named *Casebuilder*. *Casebuilder* would use the same generic framework as the other applications (JAVA JEE/Javascript/AJAX), but be made more robust and with more processing done on the client side. Using such a configuration also allowed us to easily and seamlessly work with the data within the *SIMULATE* packages.

While looking for an internship to satisfy the Capstone portion of my BS degree, I was luckily enough to find this opportunity within the university to work with Dr. Robbins. While I had experience doing Web Design and was eager to apply what I had learned, it was my interview with Dr. Robbins that led me to accept the position.

Working on this project with Dr. Robbins has been a rewarding experience. While his passion for his research was apparent with every discussion, he allowed me to find my own path and worked with me to help discover the best way apply my skills. By establishing clear boundaries and always making himself available he allowed me to work in a way that always let me feel in control, but never unsupervised.

Dr. Robbins has an especially unique ability to help guide a group of inexperienced individuals towards a common goal without sacrificing neither the tenets of the goal nor treading upon contributions of each member. While I enjoyed the many technical details of working on the project, and the countless discussions we had bouncing ideas off of each other, this aspect of his leadership is the one I hope to hold onto most and be able to emulate or at least integrate when I one day assume such a role.

Sincerely,

A handwritten signature in black ink, featuring a stylized initial 'J' followed by a long horizontal line.

Name removed.