CURRENT SYSTEM

Perspectives is a Java software framework developed by Dr. Radu Jianu at FIU that provides the means to accomplish efficient data visualization. Users are able to upload data sets and choose from a given set of viewers, compatible with the uploaded data set type. The system then presents the user with an interactive interface, called viewers.

PROBLEM

Perspective’s limitation is that it has been developed as a desktop application. The focus of this project is to translate this functionality on the web while keeping the Perspectives as the backbone. The new system relies on AJAX and JavaScript technologies to relay interaction events between the server and the client to continuously return rendered images as the visualization in a viewer.

The implementation of a web environment that enables the framework’s functionality (Upload/Delete datasets, Create viewers, Link viewers) is needed. Communication between the different pages, the controller, and Perspectives must be setup and the pages have to closely reflect the workflow used in Perspectives.

SOLUTION

To implement a web environment that enables the Perspectives framework functionality, I set up the communication between the different pages, the controller and the Perspectives framework using AJAX calls. These calls send XMLHttpRequests, asynchronously, from the client to the Controller to be handled and obtain the necessary information from the Perspectives framework, thus, enabling the required functionality. I designed the structure of the system to provide a flow similar to the framework.

REQUIREMENTS

The general requirement for the system is to allow for similar functionality that is available in the desktop version of perspectives, but made available through the web, handling most of the processing on the server and having the viewers only show images being sent continuously.

The specific requirements that relate to my role are as follows:

- Ajax communication between pages, controller, and framework
- Upload Dataset functionality
- Delete Dataset functionality
- Create Viewer functionality
- Link Viewer functionality

SYSTEM DESIGN

The online version of the Perspectives system was developed using a 3-Tier approach as shown below.

- The Presentation layer represents the pages the user can see, along with client-side JavaScript logic.
- The Business Logic layer translates the information sent from either direction, along with some additional computation.
- The Data Layer is the Perspectives framework that, among other things, provides the logic to create viewers, as well as keep track of the multiple viewers and their properties.

OBJECT DESIGN

The subsystems I was focused on were the Dataset Subsystem and the Viewers Subsystem, shown below.

The data class on the bottom-left is where the calls to upload/delete a dataset are made. The logic in the Ajax.js handles the placement of the calls to the Uploads and Controller and will obtain an updated list of available datasets after every call.

The Viewer class on the bottom-left is where the call to launch/create a viewer is made. The logic in the Ajax.js handles the placement of the calls to the Controller which interprets the parameters of the call and obtains the necessary information from the Perspectives framework.

IMPLEMENTATION

The system was implemented using Java, JavaScript, HTML and AJAX technologies.

Below is a snippet of the method that creates the XMLHttpRequest:

```javascript
function makeRequest(method) {
    var request = new XMLHttpRequest();
    if (method == "GET") {
        request.open("GET", filename, true);
        request.send();
    } else if (method == "POST") {
        request.open("POST", filename, true);
        request.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        request.send(data);
    } else {
        request.open(method, filename, true);
        request.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        request.send(data);
    }
    return request;
}
```

Below is a snippet of the method in the controller that receives the request and analyzes the request parameters and dictates how to handle each request:

```java
package web.perspectives.controller;

public class Controller {
    public void handleRequest(HttpServletRequest request, HttpServletResponse response) {
        String method = request.getMethod();
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        switch (method) {
            case "GET":
                String param = request.getParameter("category");
                String param2 = request.getParameter("page");

            case "POST":
                String data = request.getParameter("data");
                break;
        }
    }
}
```

SUMMARY

The new system that was created extends the Perspectives framework, developed as desktop application, by making it available online, while still having the interactive visualizations along with all of its functionality.

It relies on simple technologies available in most modern browsers in order to make it accessible to most people.

My focus was to enable the communication from the client to the Perspectives framework and implement a web environment that enables the functionality to upload/delete datasets, create viewers and link viewers.

ACKNOWLEDGEMENT

I am thankful for the help that I received from my group members, Santiago Pintos and Erik Franco, as well as for the guidance provided by Mershack Okoe.