Climate Visualization For Natural Resources

Final Report

Center for Sustaining Agriculture and Natural Resources

Mentor:

Dr. Kirti Rajagopalan

Gemini

Ryan Torelli, Matt Bourland, Roosevelt Young, Hasnain Mazhar

CptS 423 Software Design Project II

Spring 2018

Instructor: Aaron Crandall

TABLE OF CONTENTS

I. Introduction

The Center for Sustaining Agriculture and Natural Resources (CSANR) [1] has requested a web-based tool for aiding sustainable practices of agriculture in the face of climate change. The primary user group is agricultural professionals of the United States who advise agricultural producers on crop and livestock selection, growing and rearing practices, and pest management. In 2016, a design team delivered one feature toward development of the decision tool. During the 2017-18 academic year, a second design team is undertaking development of an additional feature.

This design project aims to build a feature that shows through data visualization how climate change impacts land used for cattle production. The feature comprises a map of the United States and a series of graphs. The map visualizes data on a large spatial scale and the graphs visualize data on a smaller, aggregated spatial scale.

II. System Requirements Specification

II.1. Project Stakeholders

The stakeholders considered in this project are livestock professionals in the western United States, the United States Department of Agriculture, politicians, and anyone concerned with the future of cattle production and related resources in the United States. The stakeholders need an easy-to-use interface for viewing and understanding complex data on climate and natural resources of their region. Our client seeks an addition to the tool that looks and feels like the existing tool. This will be achieved by using the same API (Leaflet) used in the climate visualization tool, using the same color schemes, and presenting the data in a similar way.

II.2. Use Cases

Use Case Name	Seeking general information on general area
Participating Actor	Citizen
Flow of Events	1. User selects county of interest 3. User selects indicators and climate models they are concerned with or can view all 2. Software directs user to another page where they are presented with information for the selected location
Entry Condition	User must select specific area to view information.
Related Requirements	Indicators, Climate Models, Map, Module Data View Page

Use Case Name	Seeking information on specific area
Participating Actor	Livestock Professional
Flow of Events	1. User selects data point or county that they are concerned with. 2. User may select indicators and climate models they are concerned with or view all. 3. Software directs user to another page where they are presented with desired information.
Entry Condition	User must select specific data point or county to view information.
Related Requirements	Indicators, Climate Models, Map, Module Data View Page

Use Case Name	Seeking general information on specific area
Participating Actor	Politician
Flow of Events	1. User selects county or congressional district of interest 3. User selects indicators and climate models they are concerned with or can view all. 2. Software directs user to another page where they are presented with information for the selected location.
Entry Condition	User must select specific area to view information.
Related Requirements	Indicators, Climate Models, Map, Module Data View Page

II.3. Functional Requirements

Rangelands shall orient the user to region of interest with a map on the splash page, provide data categories to query on a drop-down menu, and display responses to queries on multiple data view pages.

II.3.1. Module: Splash page

Map: The splash page shall display a centered map of the mainland United States
- Source: The client requested that Rangelands display a map of the United States. Other features display a map and the client wants consistency among features.
- Priority: Level 1. This requirement is essential.

Zoom: The map shall zoom in and zoom out
- Source: The client requested that Rangelands support zoom like other features.
- Priority: Level 2. This requirement is desirable.

Data Points: The map shall display discrete data points as pins
- Source: The client requested that Rangelands display mapped data using a similar convention as other features.
- Priority: Level 2. This requirement is desirable.

Color-coded Data: The data shall be coded by color to indicate magnitude of values
- Source: The client requested that Rangelands code mapped data values by color.
- Priority: Level 2. This requirement is desirable.

Legend: The map shall have a legend
- Source: The client requested that the map have a legend to clearly express the data color scheme.
- Priority: Level 2. This requirement is desirable.

II.3.2. Module: Drop-down menu

Indicators: The drop-down menu shall include options for the user to select climate change indicators
- Source: The client requested that Rangelands have a drop-down option for the user to select indicator.
- Priority: Level 1. This requirement is essential.

Boundaries: The drop-down menu shall include an option to show county or congressional boundaries.
- Source: The client requested that Rangelands have a drop-down option for the user to select map boundary. It is important to some of our stakeholders (politicians) to be able to get information by county or congressional district.
- Priority: Level 2. This requirement is desirable.

Overlay: The drop-down menu shall include an option to change the overlay on the map (satellite/map/hybrid).

- Source: The client requested that Rangelands have a drop-down option for the user to select map overlay. Supporting different overlays is desirable in satisfying different users. For example, an agriculture professional might want to see the satellite view while a politician might want another view.
- Priority: Level 3. This requirement is extra.

Climate Models: The drop-down menu shall include an option to select from among multiple climate models
- Source: The client requested that Rangelands have a drop-down option for the user to select climate model. There are up to four different climate models. Some users might want to view one model and others several.
- Priority: Level 1. This requirement is essential.

Time Period: The drop-down menu shall include an option to modify the time period of concern for a climate model.
- Source: The client requested that Rangelands have a drop-down option for the user to select time period. The time period of concern is a significant source of information for the stakeholders.
- Priority: Level 2. This requirement is desirable.

II.3.3. Module: Data view page

Net Primary Production: The data view page shall display information on the net primary production climate change indicator.
- Source: The client requested that Rangelands display net primary production. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Interannual Forage Variability: The data view page shall display information on the interannual forage variability climate change indicator.
- Source: The client requested that Rangelands display forage variability. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Heat Stress: The data view page shall display information on the heat stress climate change indicator.
- Source: The client requested that Rangelands display physiologic heat stress on cattle. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Relative Fraction of Woody vs Herbaceous Plants: The data view page shall display information on the relative fraction of woody plants as compared to herbaceous plants climate change indicator.
- Source: The client requested that Rangelands display inedible to edible forage. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Aggregate: The data view page shall display information on an aggregate of all the climate change indicators.

III. Source: The client requested that Rangelands display an average of all indicators. While some of our stakeholders may be concerned with only one or two climate change indicators, many will be concerned with all of them. The aggregate is a great tool to get a look at all of them together and get a feel for the big picture.

IV. Priority: Level 1. This requirement is essential.

II.4. Non-Functional Requirements

II.4.1. Quality Requirements

User help: The client requested that the design team produce a user guide in the form of a video

Browser compatibility: The client requested that Rangelands be consistent among the current updates of the three most popular web browsers: Chrome, Safari, and Internet Explorer

Scaling: The client requested that visualizations scale to screen size

Resolution: The client requested that visualizations display without pixelation at 1280 x 720 resolution

Colors and font: The client requested that color schemes and font be consistent from page to page and most closely match other features

Indicator descriptions: The client requested that descriptions of climate indicators appear on data view pages

Performance: The client requested that load times of data view pages not exceed 30 seconds

Storage: The client has acknowledged that there is a tradeoff between performance and storage and requested that storage not exceed provided data size by more than a factor of three

II.4.2. Implementation Constraints

Shiny: The client requested that web application development use this open-source R package

Leaflet: The client requested that interactive maps use this open-source JavaScript library

File formats: The client requested that data files be easily converted between GeoTIFF and CSV formats

License: The client requested that Rangelands be freely used, modified, and shared

V. Software Design

V.1. Architecture Design

V.1.1. Overview

We adopt a three-tier architecture pattern with tiers for user interface, system functionality and system storage. The user interface is represented by our ui.R file which sets up components with which the user views and interacts. The application logic is represented by our server.R file. This file reacts to user interaction, such as drop-down menu selection. The storage tier is represented by a collection of flat data files, where each file contains data values mapped to coordinates in the United States over 10 decades for a single environmental factor.

This three-tier pattern matches the framework of the R language library, Shiny, which we use as required by our sponsor. In that framework, user interface and system functionality are contained in separate files. The user interface file builds objects on screen to which the system functionality file reacts. For example, when a user makes a selection from a menu object, an event is signaled to the system functionality layer, which takes some action.

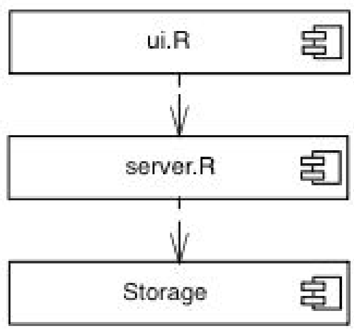

Figure 1. Block diagram representing the architectural design of our system.

V.1.2. Subsystem Decomposition

The system comprises the Client, Server, and Storage subsystems.

Client
The Client subsystem is responsible for displaying an interactive map, plots, and user input menu. The interface provides a drop-down menu for selection of climate model, map overlay, map boundary, time period, and climate indicator. The interactive map uses the Leaflet package in R. It has has different overlays (satellite, topographic, or basic) and boundaries (state, county, or congressional district) from which the user can select. As a cursor hovers over a bounded area, a pop-up displays the area's label. If the cursor selects a bounded area by clicking, a window appears containing a plot of data aggregated from the bounded area. The plot displays a mean value and standard deviation of one climate indicator forecast over 10 decades.

Server
The Server subsystem is the logic layer. It receives input from the user in the Client subsystem and requests data from from the Storage subsystem. The Server processes data and outputs

maps and plots to the Client for display to the user. The Client handles selection of parameters to which the Server responds by selecting data from Storage and generating an image for display by the Client.

Storage
The Storage subsystem is responsible for providing data to the Server subsystem. It is a collection of files that encode data values for climate indicators mapped to coordinates in the United States over 10 decades.

V.1.2.1. Client

a) Description: The Client subsystem is a front end for users to initiate use cases.

<u>Services Provided:</u>

1. Service name: setIndicator, setBoundary, setTimePeriod, setClimateModel
   Service provided to:
   Description: The service sets a menu value selected by the user.
2. Service name: setZoom
   Service provided to:
   Description: The service sets a zoom level selected by the user.
3. Service name: setCoordinate
   Service provided to:
   Description: The service sets a map coordinate selected by the user.
4. Service name: getIndicator, getBoundary, getTimePeriod, getClimateModel
   Service provided to: Server
   Description: The service gets a menu value.
5. Service name: getZoom
   Service provided to: Server
   Description: The service gets a zoom level.
6. Service name: getCoordinate
   Service provided to: Server
   Description: The service gets a map coordinate.
7. Service name: displayToScreen
   Service provided to: Server
   Description: The service outputs images to screen.

<u>Services Required:</u>
The Server subsystem renders images for display by Client.

V.1.2.2. Server

a) Description: The Server subsystem reacts to user interaction. It applies logic to map or plot data.

Services Provided:

1. Service name: identifyArea
   Service provided to: Client
   Description: The service outputs a code corresponding to a bounded area from a map coordinate. The service takes in boundary and coordinate.
2. Service name: aggregateData
   Service provided to: Client
   Description: The service outputs the average value of data for a bounded area. The service takes in indicator, boundary, time period, climate model, and bounded area code.
3. Service name: renderMap
   Service provided to: Client
   Description: The service outputs a map of the United States that is populated with color-coded data. The service takes in indicator, boundary, time period, climate model and zoom level.
4. Service name: renderPlot
   Service provided to: Client
   Description: The service outputs a plot of time-series data for a bounded area. The service takes in indicator, boundary, time period, climate model, and bounded area code.

Services Required:
The Storage subsystem provides data on which Server applies logic.

V.1.2.3. Storage

a) Description: The Storage subsystem is a back end responsible for access to data. There will be use of flat files instead of a database management system. This obviates the need to configure or manage a database.

Services Provided:

1. Service name: selectData
   Service provided to: Server
   Description: The service outputs data for a bounded area. The service takes in indicator, boundary, time period, climate model, and bounded area code.

Services Required: None

V.2. Data design

Data from file is interpreted by the R language in data frames, a data structure that binds vectors of equal length. Each file represents 10 decades of values mapped to coordinates across the United States for one climate indicator. A given file contains a data frame for each decade.

Data is selected from file spatially and temporally. For generating images to overlay on a map, values from file corresponding to a selected indicator and decade are coded to a color scheme and mapped to locations specified by file. For generating plots for a bounded area, values are subset by taking the intersection of all values in file with a spatial file that defines the polygonal boundary of the selected area.

V.3. User Interface Design

The user opens a web browser and navigates to the web address,

http://agclimatetools.cahnrs.wsu.edu/cbcct/

The user clicks on a “Rangelands” tab to view the application. The user interface is intuitive with similarity to other prevalent map web applications, such as google maps. The user can click and drag on the map to navigate to different areas and use the scroll function to zoom in and out. To visualize data overlain on a map of the United States, the user interacts with a drop-down menu, setting parameters of time period, indicator, and boundary, prompting display of mapped and color-coded data. To visualize data specific to a region, the user selects a bounded area, prompting display of a plot that forecasts mean values of an indicator.

VI. Test Case Specifications and Results

VI.1. Test Plans Overview

VI.1.1. Overview

The main objective in our testing plan is to have system code pass all of the testing activities detailed below. Some of these tests focus on the code, such as unit testing and integration testing. Other tests focus on the functionality and behavior of the code, such as system testing, performance testing, and user acceptance testing. The resources required to complete these activities are access to our code base as well as the RStudio integrated development environment. The testing activities will be conducted in order from smallest to largest units, followed by functional testing. The products that will be delivered for testing are the driver files for unit and integration testing and brief write-ups for other testing activities.

VI.1.2. Test Plans

VI.1.2.1. Completed Tests

Functional testing

The goal of functional testing was to identify the most relevant use cases and testing those with the highest probability of failure. We tested menu options for each boundary type - state, county, or district - and selected at random five bounded regions. Selection of a bounded region generates a plot corresponding to all forecast data aggregated for that region. We verified that both the aggregated data and plot matched the data selected.

User Acceptance Testing

We tested for user acceptance by confirming that the application meets every requirement specified by the client. We allowed the client to use the application and provide feedback for revision. We maintained a checklist of client requirements that we used to test item by item. After the application passed the checklist, we notified the client, made the application available, and scheduled a meeting for discussion of feedback. We incorporated feedback as a set of final revisions and delivered the application with documentation and a user help video to a predetermined web address.

VI.1.2.2. Should Be Tested

Unit Testing

To conduct unit testing on the system, we will start by breaking down the code into the smallest units possible. These units are individual functions. We will write test drivers to call each function and compare test output to expected output. If they match, the test passes. If a test fails, we will modify the function or use a different approach to obtain the desired functional outcome. Completing unit testing will increase our level of confidence in the functionality of the system. The test driver will be an R file.

Integration Testing

To conduct integration testing, we will take code units and test the interaction between them. We will test code units using as input the output of another unit. Similar to unit testing, we will check to make sure that the expected output matches the test output. If they match, the test passes. If a test fails, we will modify the function or use a different approach to obtain the output we need. Completing integration testing will increase our level of confidence in the functionality of our system. The test driver will be an R file.

Performance testing

The goal of performance testing is defining measures of performance and taking measurements under a variety of conditions. We will record and analyze the application’s response time in low, normal, and high stress situations. We will use a load test tool included in the Shiny library package, which records an application’s response time under different conditions of real-world use and server settings. Using the tool will help us record user interactions and number of concurrent users for a given duration, and playback results. If the results from performance testing are within bounds for performance requirements, we will have ensured client acceptance with respect to performance.

Usability testing

A major testing component incorporated in the checklist of client requirements is usability testing. The client has recruited a group of researchers and agricultural professionals to participate in testing a series of use cases. A protocol will be developed that standardizes three tasks. Each task will be described completely; questions will be added to each task to evaluate user understanding and task success. Some participants will take part in testing the web application remotely while others will be observed directly. The results of usability testing will be reviewed with the client for preparation of a set of initial revisions. The opportunity for client testing and feedback will follow the completion of revisions from usability testing.

VI.2. Test Results

Everything that was tested works how we expect it to and the application runs flawlessly. The code coverage of our unit and integration testing cannot be determined because those tests have not been conducted yet. Confidence in the reliability of the system can be improved by completing all of the tests in the section “Should be Tested.”

VII. Description of Final Prototype

The final prototype is a complete implementation of the product described in this document. Users can find the tool at

http://agclimatetools.cahnrs.wsu.edu/cbcct/.

When the application starts it is in the “Home” tab and displays information about the project, the development team, and instructions on how to use it. Users can find a video tutorial of how to use the tool under the “How to use” tab on the home page. To access the tool, users must click the “Tool” tab. This starts the user at the splash page, where they can the choose climate scenarios, indicators, and overlays that they prefer. Now the user can select an area of interest and be directed to a data plot for viewing. Users may choose to change the inputs on the data view page as well.

VIII. Conclusions and Future Work

VIII.1. Limitations and Recommendations

Our current prototype is for the most part complete, so there are not many limitations. One recommendation for a way to improve the tool is improve the response time of the system when fetching data and creating a graph. This would improve user experience significantly and increase confidence in the tool.

VIII.2. Future Work

The first thing that should be done to improve the system is to test it according to the specifications listed in section VI: Test Case Specifications and Results. This would ensure that every part of the system works as it should and will increase confidence in the system. To extend this project in the future, it could be integrated with work from previous senior design CSANR teams to create a multi-focus climate change visualization tool. The data visualization page could also be expanded to include more detailed data plots and a way for users to retrieve raw data.

IX. Acknowledgements

This work is funded by the Washington State University's (WSU) Center for Sustaining Agriculture and Natural Resources(CSANR). We would like to thank Dr. Kirti Rajagopalan and Dr. Aaron Crandall for their guidance. We should also like to thank Dr. Matt Reeves of the United States Forest Service for providing the climate projections data.

X. References

[1] CSANR. (2017). *Center for Sustaining Agriculture and Natural Resources* [Online]. Available: http://csanr.wsu.edu

[2] M. C. Reeves, K. E. Bagne, J. Tanaka, "Potential Climate Change Impacts on Four Biophysical Indicators of Cattle Production from Western US Rangelands," *Rangeland Ecology & Management*, vol. 70, pp.529-539.

XI. Appendix A – Team Information

Team Members: Matt Bourland, Roosevelt Young, Ryan Torelli, Hasnain Mazhar.

XII. Appendix B - Project Management

Each week we would meet for 1 hour with our mentor, Dr. Kirti Rajagopalan. Some weeks this meeting would also include Dr. Matt Reeves via Skype. The purpose of these meetings was to revise the previous week's work, identify the next week's work, and clarify the direction that our project was going. We would also have a 0.5 hour meeting with our instructor, Dr. Aaron Crandall. This meeting was to ensure progress and documentation of work. The team activities that were the most beneficial to our team were the weekly meetings with our mentor. These meetings ensured that we were focused and efficient in our work. The tools that we used to assist us were email, Slack, appear.in, and GitHub.