Climate Visualization for Natural Resources

Solution Approach

Center for Sustaining Agriculture and Natural Resources

Mentors

Keyvan Malek, Giridhar Manoharan, and Kirti Rajagopalan

Gemini

Matt Bourland, Hasnain Mazhar, Ryan Torelli and Roosevelt Young

CptS 421 Software Design Project I

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.

The data contains values for a set of environmental factors over decades at locations across the United States. The values range over time from the past to the future, where future values have been forecast by computational models of climate. The data was curated by the US Department of Agriculture, Forest Service, and has been reported in a peer-reviewed publication [2].

The outcome of the design project is delivery of one feature for extension of a web-based decision tool. The feature, hereafter called Rangelands, visualizes climate change on lands in the United States and will be made available to the public at no cost.

This document outlines an approach to the design of Rangelands. The design team elucidates a strategy by which to achieve delivery of Rangelands in the following eight sections:

Section II describes the function and design of Rangelands
Section III proposes a system architecture with component breakout
Section IV presents data structures and databases
Section V shows the user interface by use case
Section VI lists achievements in progressing toward the project outcome
Section VII shows a work schedule in the form of a Gantt chart
Section VIII presents a glossary of specialized terms
Section IX presents references

II. System Overview

This design project aims to build the Rangelands feature as a component of a web-based tool that aids planning in agriculture. Rangelands informs and educates users on the effects of climate change on the environment. It draws on climate models to produce maps and graphs that visualize environmental variables in time and space.

Decouple data. Rangelands shall incorporate design choices that disassociate data and code. With minimal code modification, Rangelands shall evolve as data evolves. Rangelands shall accommodate data updates and changes to data parameters such as number of climate models, indicators, and time periods.

Decouple spatial boundaries. Rangelands aggregates data by boundary selection for graphical display of indicators. One boundary option is congressional district, which is subject to change every decade. Rangelands shall aggregate data by congressional district with the foresight that district boundaries change.

Performance. Rangelands operates on large data volume, which slows computations and display of visualizations. Where possible, computations or visualizations shall be preprocessed to enhance performance.

Browser compatibility. Rangelands is a web application displayed by a browser on a screen. As browsers update, Rangelands shall continue to provide a consistent user experience. As screen resolutions increase in density, Rangelands shall maintain the quality of its visualizations.

The design team shall make choices that enable Rangelands to visualize climate for years to come with minimal code modification.

III. Architecture Design

III.1. Overview

The pattern we adopted in our software is the three-tier architecture pattern. This pattern fits our system well because of the defined format from the Leaflet API, which we are required to use by our sponsor. There are three major components that we must be concerned with in the three-tier system: user interface, system functionality and storage system. This is the exact way that Leaflet works so I assume that the developers followed this pattern when creating the API. The user interface layer is represented by our ui.R file. This file sets up user interface components and basic components that user views and interacts with. The application logic layer is represented by our server.R file. This file controls how the tool works. It handles how to react when the user interacts with the tool (clicking, scrolling, etc). The storage layer is represented by our storage system. It is responsible for the storage, retrieval, and query of objects. Currently this does not exist in our prototype because we are still waiting on data from our sponsor.

There are some framework and data constraints that exist in this project. The client requested that we develop the application using Shiny, an open-source R package, and Leaflet, an interactive map library. The client also requested that data files be easily converted between GeoTIFF and CSV formats.

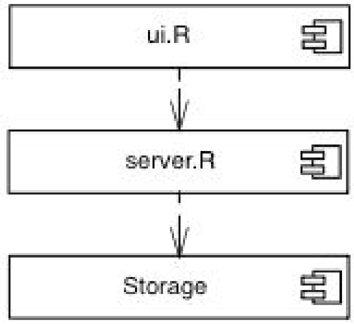


Figure 1. Block diagram representing the architectural design of our system.

III.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. Boundary overlay currently has a clicking functionality that shows the name of the area. In the future, clicking on an area of the map will generate plots of climate indicator over a time period.

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. Currently, the Server processes data and populates a map with image files. The map shows overlay, boundary, and color-coded data points according to user selection. In the future, the Server will be responsible for transformations on the data and generation of plots.

Storage
The Storage subsystem is responsible for providing data to the Server subsystem. It has not been implemented.

I.1.1. Client

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

Services Provided:

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.

I.2.1 Server

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

<u>Services Provided:</u>

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.

I.3.1 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

IV. Data design

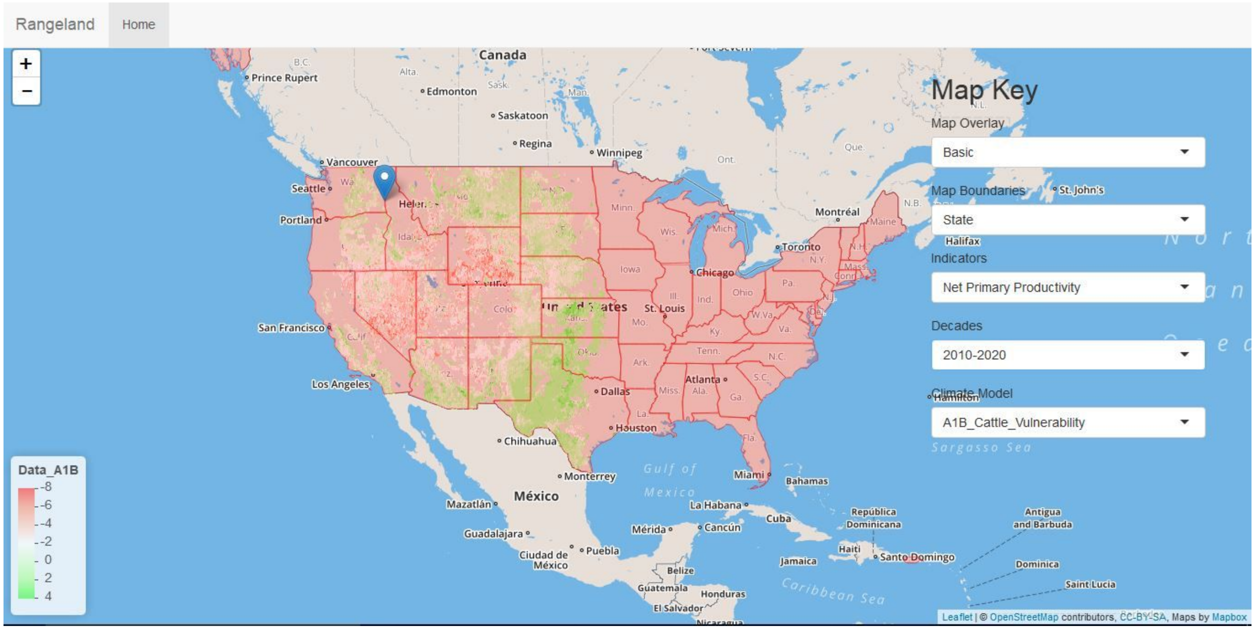
Data structure is abstracted by the Shiny framework. The selection and manipulation of data is performed without specifying data structure.

V. User Interface Design

The user will open a web browser and navigate to the web address,

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

The user will click on a “Rangelands” tab to view the application. Our user interface is intuitive and will feel very similar to other map web applications you may have used, such as google maps. You can click and drag on the map to navigate to different areas and use the scroll function on your mouse to zoom in and out. To access data, use the drop down menus on the right side on the page to set your preferences (time period, indicators, boundaries) then click on the map for your desired location. All of the use cases in the Requirements Specification document will utilize these interfaces for user interaction.



VI. Summary of the State of This Project

Currently, the majority of the front-end GUI has been implemented and the foundation of the back-end has been put in place. Most aesthetic features of the program exist; however they still need to be tweaked and optimized to run on a restricted-resources interface. Nonetheless, much has been accomplished, including:

- Implementation of an interactive map
  - Display of map overlays
  - Display of area boundaries
  - Display of color-coded data points
- Implementation of plots
  - Display of fixed time series data
- Determination of size of storage space

The project is moving along and will soon be in a prototype stage to better solicit client feedback.

VII. Future Work for This Semester

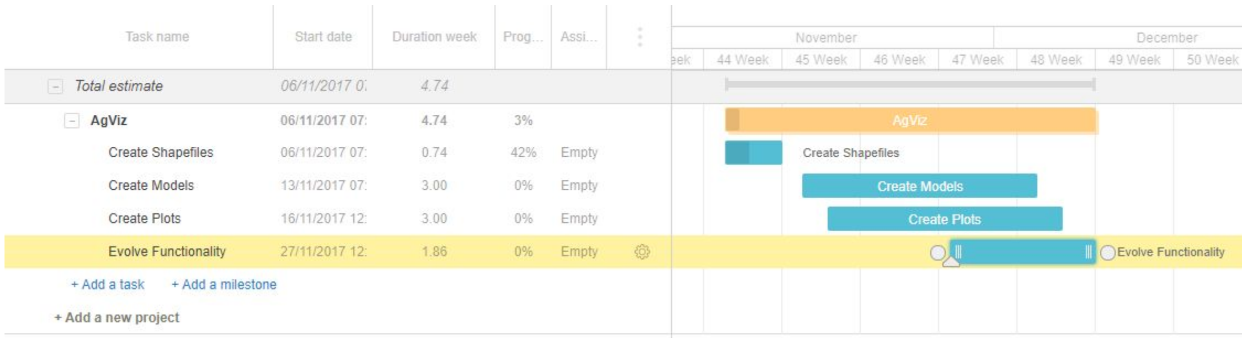


Figure 2. Tabular work schedule and Gantt chart.

The remaining work for this semester consists of modifying the implementation with respect to boundaries and plots as well as gaining a better understanding of the dataset. This dataset will be used to show change in climate indicators from present day through year 2100.

The tasks underway in need of completion by this semester's end are the following:

- Cropping shapefiles: The shapefiles representing boundary by county, state, and congressional district need to be cropped to exclude areas that do participate in livestock production.
- Creating plots: The design team is exploring ways to create useful plot visualizations for different use cases.
- Understanding the dataset: The design team needs to understand the dataset better in order to extract useful information and produce insightful visualizations.
- Evolving functionality: The design team is improving the code base as we come to know the dataset and capabilities of R, Shiny, and Leaflet.

Creation of the semester alpha prototype is fairly completed in relation to how the project was envisioned to go. Heading forward into the next few weeks of the semester the AgViz team will focus on implementing the features plot and models and focus on improving the current functionality and quality of code in the current project by implementing bug fixes and code revisions.

VIII.Glossary

Agriculture Professional - Crop consultant who advises on crop choices or university extension staff member that advises on best practices

Net Primary Productivity - The amount of carbon uptake after subtracting Plant Respiration (RES) from Gross Primary Productivity (GPP). GPP is the total rate at which the ecosystem capture and store carbon as plant biomass, for a given length of time

Interannual Variability - Value describing the interannual standard deviation in annual average forage quantity for a region

Heat Stress - Negative effects on cattle due to higher than usual temperatures

Relative Fraction of Woody vs Herbaceous Plants - Ratio of inedible to edible plants

Leaflet - Open-source JavaScript library for interactive maps

Rangeland - Open country used for grazing or hunting animals

Shiny - An open-source R package that provides an elegant and powerful web framework for building web applications using R

IX. 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.