Visualization of Earth Modeling Systems (VEMS)

May1701

Client: Dr. Chaoqun (Crystal) Lu

Faculty Advisor: Dr. Johnny Wong

Team:

Kellen Johnson - Communication Lead

Anish Kunduru - Team Leader

Eli Devine - Webmaster

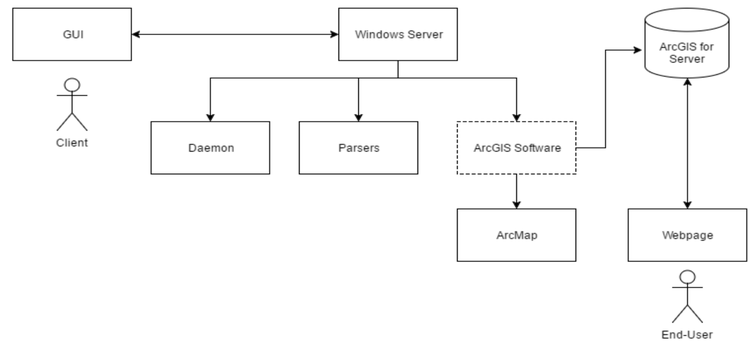
Julio Salinas - Concept Holder

Project Statement: Professor Chaoqun Lu creates historical and projected estimates of ecosystem functions through an Earth Modeling System. In order to build a bridge directly between stakeholders and modelers, our client wishes to visually display time-series spatially explicit model output data across regional to global scales in decadal or centurial time periods. Professor Chaoqun Lu would like an interactive method to display Earth system models in lieu of standard, non-interactive, image files.

Solution: Create a system to automatically convert the client's datasets into 3D maps which allow users to view the regional or global model output data (daily, monthly, annually) in an interactive space.

Design Approach:

Block Diagram:



Modules:

GUI:
- Allows Client to interact with Daemon

Daemon:
- Error handling
- Ensures placement of input files within the server
- Handles Client requests

Parsers:
- Translates Client data into ArcGIS readable format (CSV)
- Transforms CSV into spatial geodatabase
- Places data source into map document
- Publishes map services to ArcGIS for Server

Website:
- Loads requested maps via RESTful JavaScript API

Intended Users:

- Students
- Professors
- Policymakers
- Land Managers
- Grant Foundations

Intended Use:

Educate the public on the effect of anthropogenic stresses placed upon the environment. Ultimately, this will increase awareness of climate change and assist policymakers and key stakeholders in related decision making.

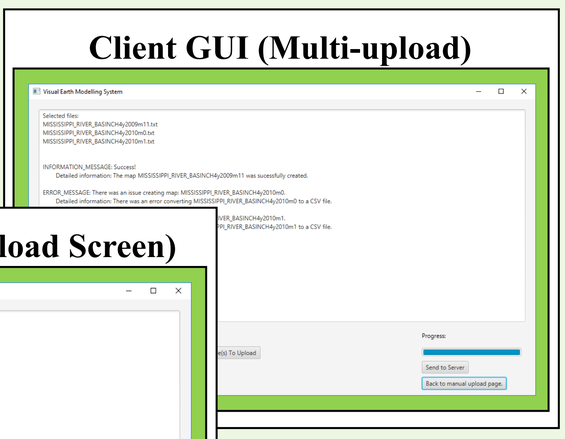
Functional Requirements:

- The product shall allow the client to upload raw ASCII data to the server.
- The product shall parse all uploaded data from an ASCII file into a format readable by ArcGIS.
- The product shall automate the steps traditionally taken to create a map-service (via ArcMap).
- The product shall allow end-users to dynamically select and view any generated map.

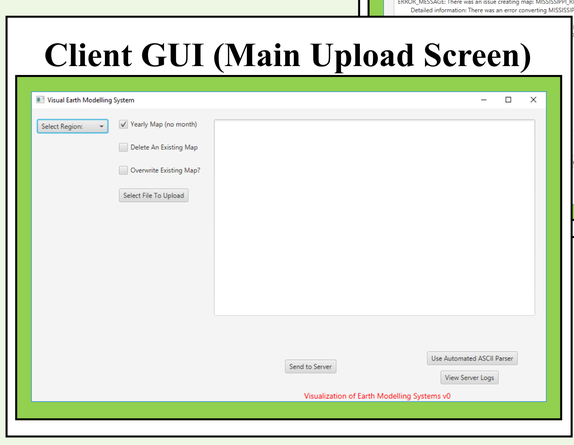
Non-Functional Requirements:

- Parsing of dataset within 15 seconds.
- Map generation within 10 minutes.
- Map data should be distinguishable by color and/or size.
- Maps services should load in less than 30 seconds.
- 8 Simultaneous users (across platform).
- 2 Simultaneous users (same map service).

Client GUI (Multi-upload)



Client GUI (Main Upload Screen)



Testing:

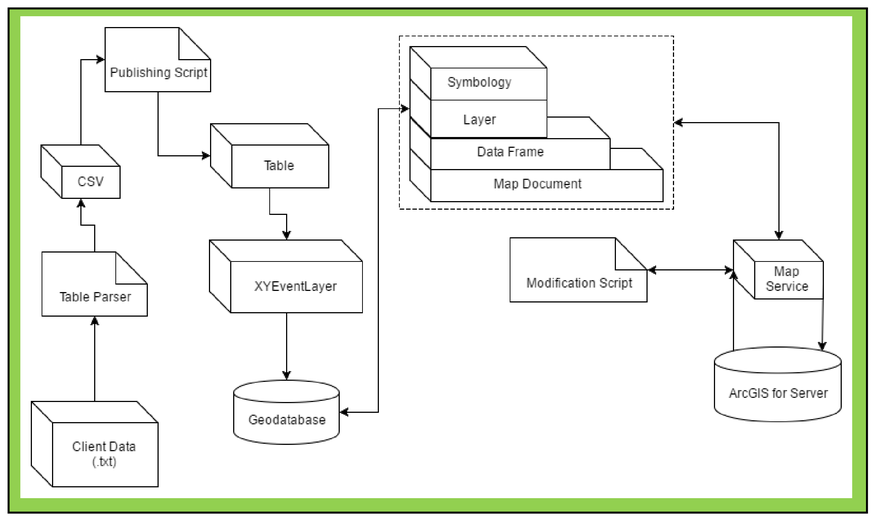
End-User Reactivity:
- Load Time (Rendering)
- Interaction Time (Zooming & Panning)

Input Data Validity:
- Visual Confirmation

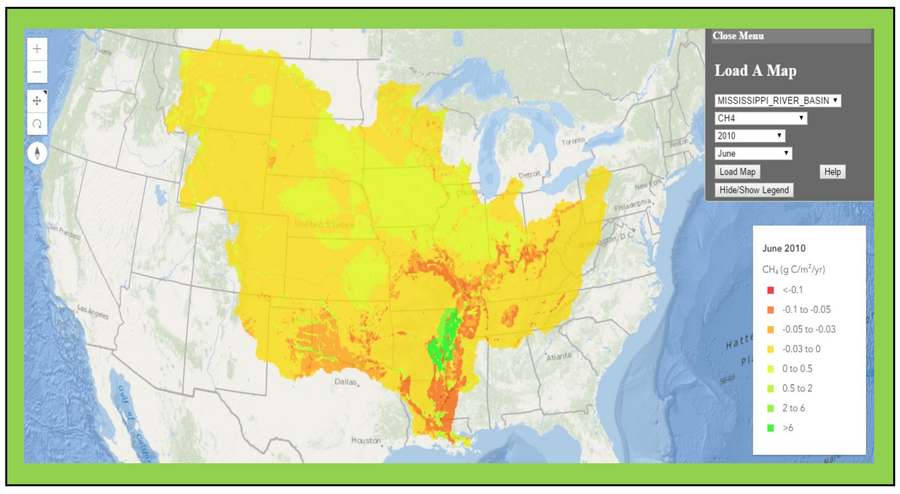
GUI:
- Log errors
- Notification-based interaction

Resource Management:
- CPU & RAM usage upon peak end-user utilization of map services

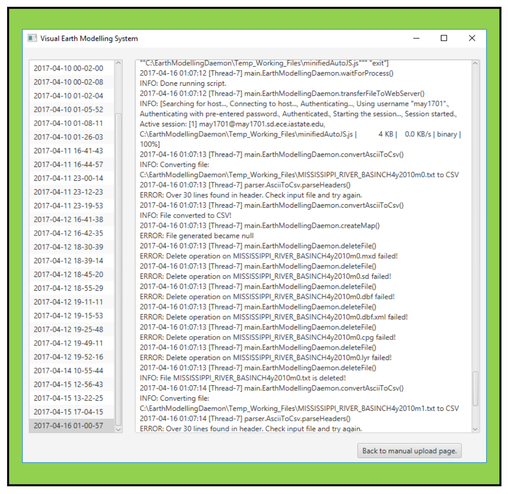
Data Transformation Flow Diagram



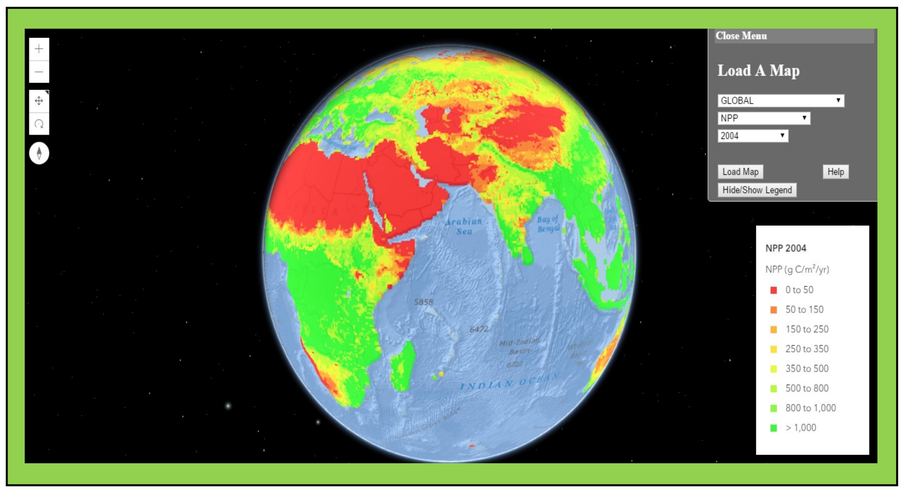
End-User Website (Monthly Map)



Client GUI (Server Logs)



End-User Website (Global Map)



Programming Languages:

- Automated Daemon: Java
- Data Transformation Parsers: Java & Python
- End-user Website: HTML & JS
- Client GUI: Java (JavaFX)

ArcGIS Tools & Software:

- ArcMap: Allows for creation of template maps.
- ArcGIS for Server: Hosts generated map services and provides Python libraries (Arcpy).
- ArcGIS API for JavaScript : Provides RESTful API to load and display map services in a browser environment.

Misc. Software:

- Procrun: Installation of daemon.
- WinSCP: Transfer generated JS to webserver.
- Closure Compiler: Minify generated JavaScript.
- Tiny log: Log critical information.

Operating Environment and System Specifications:

- Website is available to anyone on the ISU intranet.
- Windows Server: O/S that all our software runs on.