

AtlasCMS: An Online Atlas Content Management System

Michael Marti

Student

Department of Computer Science

michael.marti@my.und.edu

Marshall Mattingly

Student

Department of Computer Science

marshall.p.mattingly@my.und.edu

The goals of this collaborative proposal are to: *i*) create a content management system (CMS) that will allow students from multiple disciplines to upload and maintain maps, graphs, narratives, audio, and other data representing demographic, economic, and social changes across the state of North Dakota in the past 125 years of statehood; *ii*) implement algorithms and interfaces that allow the students to represent complex maps and data sets interactively from ArcGIS, digital mapping software; and *iii*) assist with the initial procurement and setup of the web server, database, and storage for the CMS.

Scientific Merit: As North Dakota approaches its 125th anniversary of statehood, there is a wealth of demographic, economic, and social data available relating to the diverse religious, ethnic, and social groups of North Dakota. However, this data is scattered across many fields, including anthropology, history, religious studies, sociology, and American Indian studies, without any single source location to view multiple narratives from the many fields. Creating interactive maps and graphs with accompanying text and audio narratives across these disciplines will allow students and the general public to learn about multiple aspects of North Dakota, creating connections between the multiple disciplines that may not have been apparent otherwise.

Broader Impact: Creating a CMS instead of a hard-coded website allows for expandability into other projects that wish to represent geographical data sets with an interactive user experience. Utilizing an open-source license for the CMS will allow others to expand upon the features of the AtlasCMS to their specific needs, further increasing its versatility.

Approach: To allow for maximum portability and simplification of coding, we will utilize several robust, well-tested open source web application programming interfaces (APIs) and frameworks, such as JQuery and Bootstrap for the client-side interface, and Flask and Jinja for the server-side implementation. The database will be served by MySQL and will allow for the efficient categorization of chapters, narratives, maps, and other data sets. We will use an agile prototyping development methodology to create several different design and data representation mockups, picking the best aspects from each prototype based on feedback from the multidisciplinary students to create a polished end design.

A Atlas of the Historical Geography of the United States [1]

Excellent maps and a good table-of-contents that we may look at integrating, but there is no narratives associated with the maps, and it is exclusively static maps, no graphs or interactive elements.

B Bootstrap [2]

Bootstrap is a frontend framework that is used to develop responsive web-based applications specifically with mobile first projects in mind. Bootstrap at its core supports fluid layouts and multiple design views right out of the box with little to no additional coding. This makes it possible to easily design templates with fluid layouts that automatically scale and resize based on how a user is viewing the app. This also minimizes the development time required since a single template can be designed and developed, while leaving the fluid layout to be scaled automatically by Bootstrap's JavaScript plugins.

C Data-Driven Documents [3]

D3.js is another JavaScript library that allows countless ways to interact with different data sources in tables, graphs, and maps, just to name a few. D3.js utilizes the newest features in HTML5 specs, and allows the ability to have arbitrary data sources and create powerful data-driven interfaces for a unique user experience.

D DjangoCMS [4]

Django CMS is a fully customizable open-source content management system written in Python using Django. Just releasing version 3.0, Django CMS overcomes many of the CMS obstacles encountered by the project: version control, draft vs published pages, WYSIWYG editor for text, and handling add-ons, such as complex graphs. Utilizing an already existing framework that is used by businesses as well as blogs for the CMS backbone allows us to focus on creating the atlas specific add-ons, primarily interactive maps and narratives.

E Django Template System [5]

Django has a powerful templating system that allows you to mix Python code with HTML. Django's templating system allows you to create a raw HTML page, but also the ability to pass in Python objects to iterate through and create dynamic content on the fly.

F Gridster [6]

Gridster is a jQuery plugin that enables drag-and-drop grids while utilizing only JavaScript and CSS. This means not only can we take advantage of the responsiveness of Bootstrap, but we can also resize and move individual elements within the web page, or even allow the user to define their own layout based on their personal preference.

Gridster gives us the ability to even display different views within the same layout based on the size of each widget or tile. For instance, making a widget wider may display a new column of information in a table, while making a widget taller may display an image versus text.

G jQuery [7]

jQuery is a JavaScript library, which makes things like DOM traversal and manipulation, event handling, transitions, and AJAX requests easy with an API that offers excellent cross-browser support. jQuery ensures a consistent experience for users even across different browsers by implementing multiple fallbacks for a multitude of browsers. This means utilizing the newest standards for browsers that support the full HTML5 specification, while still maintaining compatibility with older browsers by falling back to older standards to maintain the same level of functionality and features.

H Python for ArcGIS [8]

Python for ArcGIS is a Python framework for interacting with ArcGIS data. This software suite gives us the potential to access and manipulate raw ArcGIS data, using a DjangoCMS application along with JavaScript to represent accurate and interactive maps from raw ArcGIS data. This would allow us to create dynamic elements without the need to pre-build several maps.

I Urban Layers: Manhattan's Urban Fabric [9]

Urban Layers is an interactive map created by Morphocode that explores the structure of Manhattan's urban fabric. The map lets you navigate through historical fragments of the borough that have been preserved and are currently embedded in its densely built environment. The rigid archipelago of building blocks has been mapped as a succession of structural episodes starting from 1765.

J Washington DC: Our Changing City [10]

Washington DC historical data in an interesting form. Integrates narratives on the left-side and a corresponding graph, map, or interactive element on the right side. Using the CMS, we will likely create something similar to this. Chapters are apparent and split the data into logical groupings.

- [1] "Atlas of the historical geography of the united states," <http://dsl.richmond.edu/historicalatlas/>, accessed: 2014-10-21.
- [2] "Bootstrap," <http://getbootstrap.com/>, accessed: 2014-10-21.
- [3] "d3js data-driven documents," <https://github.com/mbostock/d3/wiki>, accessed: 2014-10-21.
- [4] "Django CMS documentation," <http://docs.django-cms.org/en/support-3.0.x/>, accessed: 2014-10-21.
- [5] "Django template language," <https://docs.djangoproject.com/en/1.7/topics/templates/>, accessed: 2014-10-21.
- [6] "gridster.js," <http://gridster.net/>, accessed: 2014-10-21.

- [7] “jquery,” <http://api.jquery.com/>, accessed: 2014-10-21.
- [8] “Python for arcgis,” <http://resources.arcgis.com/en/communities/python/>, accessed: 2014-10-21.
- [9] “Urban Layers manhattan urban fabric,” <http://io.morphocode.com/urban-layers/>, accessed: 2014-10-21.
- [10] “Washington DC our changing city,” <http://datatools.urban.org/features/OurChangingCity/>, accessed: 2014-10-21.