Executive Summary

List of Abbreviations

Introduction

Hydrologic libraries have been used in the scientific communities to simplify the workload that a researcher most due to obtain basic and reliable results.

Statement of Problem

Purpose of the Study

* Common web browsers have used Javascript since 1990 as the first programming language that was generalized for common usage.
* Although there are several libraries on the literature and internet communities that provide similar ideas, none of them have included topic such as time series analyzes, hydrological modules for basic analyzes, and mapping structures.

Objectives

Overall, Hydrolang aims to use the technology implemented within common web browsers in order to perform several hydrological analyses separately and delivering them on object-like arrays to be used for further analytical uses. This has been done by creating a series of components within a library like framework in which the user can obtain different algorithms to calculate different processes and from there obtain the aforementioned objects. The main structure of the Hydrolang is described in more details on the methodology section of this thesis.

The components defined and created within the library follow the next structure:

* Data
* Analyzes
* Mapping
* Visualization

The data module has been written so that it can retrieve information from sources that provide data and depend specifically on the type and amount of data required. The data that has been obtained is later processed through another function contained in the module which transforms it into common files that can be processed by common language such as JSON, XML and CSV.

The analytical module contains two different main functions: stats and hydro. Stats function has different tools to process data and performs basic statistical analyzes to understand and clean up data. The hydro function contains tools that depend on the type of data and analysis required. It allows the to process the data that has been obtained in the data module or previously owned by the user and obtain time series analyses.

The mapping module consist of mapping tools that allow the user to include layers of data obtained from previous modules. The layers can be flooding extents, markers or layer of analyzed data. Finally, the visualization module provides the user with charts, reports and other visual tools that aid the user to better analyze the obtained data from previous modules or previously owned.

The modules can be accessed using the chaining properties of high-level modules in javascript, and after the functions have been used, they can be saved on array-like objects.

Definition of terms

* Standardize the use of client, user, developer.

Literature review

Due to the need of providing a library that is easily accessible using web browser technology, the development was established to be done in Javascript. Since its development in the 90’s, the implementation and standardization of the language on the most commonly used web browsers and the different amounts of libraries and frameworks that deal with more responsive and adaptive web technologies create a suitable environment for the scope of the project. Specifically, dealing with a prototypical document object model which define first classes as functions as well (Punchatz, 2017).

The research question that was used to find similar approaches to what Hydrolang was to become was the following:

*“Which applications are available on the literature and web regarding open source, web based environmental, hydrologic and hydraulic analyzes?”*

In order to answer the research question previous to the development of the project, the following resources were used:

* GitHub
* Science Direct
* Google Scholar
* Research Gate

From the results, NUMBER were papers and projects that used Javascript as the main programming language used for module development, NUMBER used Java, NUMBER used Python and finally NUMBER used PHP or any other language. Overall, the papers can be categorized on the type of results that are obtained which are based mainly on environmental data, data retrieval from environmental agencies, visualizations applications that used Google Maps or other web services for deployment. The following figure summarizes the number of papers that were reviewed previous to the development of Hydrolang.

Object oriented architecture has been defined as:

“*the methodology for creating software components in the form of reusable libraries exclusively restrained to a specific domain ontology*”

Different approaches have been taken to create library based analytical tools for research and education purposes.

Design Methodology

Requirements

The idea for creating Hydrolang comes from the ease of use to run code using a client-side scripting language and utilizing only the technology implemented on the browser.

Components

The modules created on Hydrolang are aiming to provide the following

* Be able to understand, retrieve and transform data from governmental or industry sources on environmental data.
* Data comprehension, analysis and

Architecture

Using a library-oriented architecture, Hydrolang has been developed to explode the functionalities that the dynamic coding style of Javascript along with taking advantage of the principles and methodologies of the ontology domain of reusable software libraries. By establishing the latter, functions are accessed using inheritance from classes as established on ECMAScript 6, in which the syntactic sugar has been reduced and chainage is achieved through the usage of basic dot notation (Engelschall, 2017). Instantiation was not considered because by using classes, a cleaner and responsive code was obtained; thus, a decision was made to work within the library creating code wrapping of objects to access the functionalities on lower levels. This was achieved by implementing a modular approach which allows for the protection of the execution context of the given modules and deliver to the public just was needed to expose. In general, the following is the modular tree which the library follows:

* Object mutability in JavaScript. “Objects inherit from objects. What could be more object oriented than that?”
* Inheritance from classes has been used using the ECMA6 version, instead of creating prototype which would require for doing instantiation for each function that is required to be used.
* Object initialization using object or array is also used within the library for the creation of new objects that can be used by the client.
* Functions are considered objects.
* Inheritance in JavaScript is covered by delegation automatism bound the the prototype property of constructor functions.
* Important to mention within the review for javascript development: event loop and how the stack works for working with different functions within a class.

External libraries

To achieve a cleaner and more up to the point code, different libraries have been used to within the Hydrolang framework to make the code more readable and easier to follow up from the user point of view. The main libraries used were the following

* Underscore.js
* JQuery.js
* Ajax.js
* Google

JQuery is a javascript library that is used to create more concise code writing by enabling shortcuts on javascript coding and obtain same outputs as compared with pure javascript coding. It is mostly used for creating documens such as HTML and enable traverse manipulation, handling of events and implementation of features for asynchronous requests using Ajax simpler to use, as described on the library main page (JQuery, 2020).

Underscore is a library collection of over 100 functions that enable the user to use functional helps that enable quicker and reliable implementation for things such as quick indexes, deep equality testing, function creation, etc (Underscore, 2020). Functions such as

User interface

Complementing with the current technologies, Hydrolang has been designed using the most current version of Javascript coupled with HTML and CSS to create a basic and robust user interface.

Results

Components

The library was developed mainly on the interoperability of modules that enables the user to use all the functionalities from the components independently without requiring much dependency and removing possible future inheritance issues. This means that the user is able to use data retrieved using the functionalities on the library or use external files to perform statistical, hydrological or visual analyzes on data. Each component has dependencies that differ on the functions accessed within as well as the usage of different frameworks.

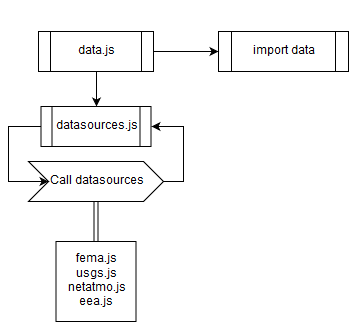
Data

Functionalities

The data component is subdivided in the data.js module which handles the data queries coming as inputs from the user; datasources.js which is used to export the source parameters from the data components and the data sources folder which contains JSON files that contain the necessary query information to retrieve data from different sources. As of August 2020, the sources available within Hydrolang are the following:

* FEMA
* USGS
* NETATMO
* EEA

The connectivity of the module described in the following diagram. It is important to notice that datasources.js was created as a library exporter only to call on the different types of sources that are available on the library.



The main module, data.js, is composed of a retrieve function using as parameters a callback function and a parameter object that is used depending on the type of data source that is to be obtained. It uses JQuery to call on the properties that the data object has and compare them to the source and see if the data is available. If it is available, it uses a get function to obtain the data on a JSON format and if it not available, then displays an error stating that “No data has been found for given specifications.”

The transform function is also included inside the data.js file, which contains commands to transform the data downloaded or any data that the user has and transform it into CSV, XML or JSON so that it can be used in further analyses.

Use example

The data module is located on the 3rd level of the chaining, right after the declaration of a new Hydrolang object from the functions in the second level. the following is an example of the implementation of the data module. Notice that the user must create an object that is adaptable to the data retrieval sources that Hydrolang supports.



The object created for retrieving data will yield a raw object file that will be on the available format that the source can provide. Simultaneously, the user can use the data transformation function to change the data into the formats that the library supports (CSV, JSON, XML, ASCII).

Analyze

Functionalities

Use example

Conclusions/Discussion

References

Ideas to explore

* What is the name of the approach which professor Demir has taking in regard of using the modules of hydrolang with the usage of chainage.
* Which are the type of basic hydrological analyzes that are commonly used by scientists to approach hydrological data.
* Depejdencies between components must be declared on top of the files that are used, using ECMAScript 6 import syntax.