

Geofront : presentation

Front

Hello, thank you all for coming! Today I would like to present the results of my thesis, Titled: Geofront: Direct accessible GIS tools using A web-based visual programming language

- This presentation serves as a showcase, I will not cover the full study, all sub-questions and tests.
- But instead, I will Show an overview the main reasons and ideas of the study, and the results of the study.
- I hope at the end of this talk, all of you have learned something new and interesting.

01 Introduction

Motivation

GIS

- The field of GIS: (Geographical Information Science) commits itself to the betterment of collecting, processing, storing, and viewing geodata: data about the earth
- By doing so, it offers the world priceless information about the land we build on, the seas we traverse, the air we breath, and the climates we inhabit.
- This information is foundational for many applications, including environmental modelling, infrastructure, urban planning, governance, navigation, the military, and agriculture.
- As such, the field of GIS is continuously looking for new ways to provide these fields and industries with both the data and tools they need to succeed.
- This thesis concerns itself with the latter: Providing tools.

TWO FORMS

- many of these tools involve software, and software can be written in one of two ways:
 - We either write software applications, or software libraries

APPLICATION

- Applications are the type of software most of you are probably familiar with,
 - this is software which can actually be 'used'
 - they offer interaction to end users
- IN GIS
 - applications are important, because they offer Visualization and analysis of geodata

LIBRARY

- Libraries among the general public are lesser known
 - Libraries can be seen as offering Re-usable tools for developers
 - They often offer more rich functionality, but cannot directly be used
- In GIS
 - The libraries are often where the "real" transformation and analysis capabilities
 - Any time an application does some form of transformation, analysis, usually these are done using the functionalities of a library

This is important to recognize

- Libraries cannot be directly used
- They can only be used within an application
- For a software developer wanting to utilize the functionality of a library, this is not a problem.

CORE PROBLEM:

- The problem this study seeks to address, is that the core transformation and analysis tools found in certain GIS software libraries, are normally not directly accessible by practitioners in the fields mentioned above: governance, infrastructure, urban planning, etc.

INDIRECTION

- The tools can only be used indirectly,
- and only when a software developer has incorporated the exact functionalities an end-user is looking for, in an application.
- Similarly, if research leads to a new GIS library, end users are at the mercy of a software developer implementing the functionalities of said library as a usable application, or as a plugin for an existing GIS environment like QGIS [QGIS Community, 2022].

SCENARIO

- This leads to the following scenario:
 - maintainers of a GIS library often find themselves in the situation of having to maintain and synchronize a great number of bindings and plugins, which limits innovation.
 - meanwhile, end users are dependent on all these different layers of developers.

MOREOVER

Moreover, even if a library is incorporated in an application, the tools are almost always less feature rich,

- Telephone game

and non-composable: The output of one procedure cannot automatically be used as input for another. This leads to labor intensive procedures and repetitive workflows, as opposed to automated, re-usable procedures.

And this is part of a wider conundrum:

- unsolved

- How can we have software which is maximally functional, composable, and usable at the same time?

SUMMARIZED

These problems are summarized as a problem of indirect access, from the point of view of end-users. Leading to disadvantages for end-users and library developers.

It is safe to say that the limited reach of these libraries translate to a reduced societal impact, and with it, the GIS research these libraries are based upon.

02 Objective

Goal

- The goal of this study is to allow GIS practitioners without a background in software development, to access the full potential of core transformation and analysis capabilities found in native GIS libraries.

APP + LIB

- An important component of this goal is to partially bridge the divide between application and library.
 - And this can be done by either adding composability and automation to applications
 - or by adding usability and a graphical user interface to libraries

HOW

- The study attempts to meet this goal by presenting and prototyping a novel method, by combining three technologies, to form a 'Web Based Visual programming language using WebAssembly'
- And, we will get to what that exactly means, and why it may provide a solution

PROPOSAL

This study attempts to

Objective

Research Questions

The objectives outlined above lead to the following main and supporting research questions:

Is a web based VPL a viable method for directly accessing native GIS libraries with a composable interface?

Supporting Questions

- What GUI features are required to facilitate this method, and to what extent does the web platform aid or hurt these features?
- To what extent does this method intent to address the discrepancies between software applications and libraries, as described by Elliott (2007)? Does it succeed in doing so?

- What are the differences between compiling a GIS library written in C++ to WebAssembly, compared to compiling a GIS library written in Rust?
- What measures are taken to make this VPL scalable to large geo-datasets, and how effective are these measures?
- How does this method compare to existing, alternative VPLs and browser-based geocomputation methods, regarding the properties relevant to the goal of direct accessibility?

03 Background

- I will briefly mention these three fields, after which I can explain how they fit together

Static Web Applications

- A component to the aforementioned goal is accessibility.
- the web may be a good platform for any project desiring a level of 'accessible'.
 - it offers distribution advantages
 - can be used without installation.
 - can be used by multiple platforms
 - Windows, Mac, Linux, Ios, Android
- STATIC web applications, meaning that it does not use an active back-end.
 - More composable (no ties to backend)
 - Cheap to host, both in terms of performance and actual hosting costs

Visual Programming

- A Visual Programming Language (VPL) is a type of programming language represented by a GUI, rather than a text-based source code. A VPL 'script' might take the shape of for example a graph (Figure 3), or a block-based instruction set.

WebAssembly

- The second techonogy I would like to bring to your attention is WebAssembly
 - which may pose an answer for this idea of 'direct accessibility'.
- WebAssembly is a binary compilation target meant for a virtual runtime.
 - This means that multiple different programming languages can 'choose' to compile to WebAssembly
 - Wasm binaries can be run in any environment and language, as long as such a runtime is implemented.
- Without going into further detail, WebAssembly has two use cases relevant to this study:

WASM USE CASE 1

WebAssembly partially mitigates the need for web-based, JavaScript alternatives of native software applications and libraries.

- WebAssembly has multiple use-cases,
- This means that it becomes a 'run anything anywhere' tool.

Such a runtime has been added to all major web browsers since late 2019 [w3c, 2019].

Among many purposes, it offers a method to run native software as (part of) a web application.

WASM USE CASE 2

Generic Library binding

04 Methodology

PROPOSED METHOD

- With those aspects explained, I can explain the proposed method.
 - It consists of two components:
- First, we will compile existing GIS libraries into WebAssembly.
 - This way,
 - we can both start using these libraries in web-browsers,
 - as well as allow them to be used as a generic interface, serving any language
- Secondly, A web-based visual programming language is required, Which is then able to access these generic bindings just like any other language.
 - then, it being a web-based, visual programming environment, it can serve the functionality within these libraries in an directly accessible, and composable manner.

OVERVIEW

- WEB VPL
 - Design and implement
- Library plugin system
 - Design and implement
- Series of tests
 - to acquire data about the quality of both

Web VPL

Plugin system

REGULAR CASE

OUR CASE

In our case we can do something different

- if in stead ot

05 Results

(show the extend to which things work)

Geofront

A custom implementation was needed to meet all requirements

- degrees of freedom in designing appropriate GUI, and framework
- Because all these systems are intertwined, control was deemed important

A novel, custom VPL is needed to fit all requirements.

A dataflow VPL must be designed for web usage, which offers a range of GUI nodes to serve as a generic GUI, and to allow applications to be composable

Effects of parameters on a certain prodecure

- informal benchmarking

Plugin system implementation

It turned out to be possible

Conclusion

Multiple observations can be made based on this demo. First, notice again how all steps mentioned correspond to the nodes of the pipeline. None of these steps are final. A different application could be created at runtime, by for example, adding a laz writer component to save the resulting subset as a point cloud itself, or by using a component to perform a solar potential analysis. This application strongly suggests that Geofront can indeed be used to directly utilize native libraries as part of a "real" use-case. Additionally, it demonstrates that Geofront allows not only composability between libraries, but also between libraries and full-scale, existing applications.

Usage

Features

06 Conclusion

Answer to Library / Application divide

- Based on the features, it can be stated that the implementation was able to address all three discrepancies to a certain extent
 - Results were reasonably successful.
1. Library capabilities get lost when used in an application: Because of the plugin system of this method, all functions which are included in a wasm compilation can be accessed by a user from within an application. Additionally, the plugin boilerplate comparison of Section 6.3 showed that the no-boilerplate setup makes it simpler to add a new library as a plugin. Both features combined may lead to less features ending up 'lost in translation'.
 2. Applications are not further composable: The implementation showed that this web-based VPL can make existing applications composable, evident from the demo application presented in Section 6.2. In this example, the Potree application was composed within a pipeline to extract a DTM or DSM, while retaining full functionality as a standalone application.
 3. A library offers no visualization or GUI by itself, and must be turned into an application before it can be used: Because the method presented in this study succeeded in combining GUI components with low-entry barrier plugins, and because it allows the usage of unaffiliated WebAssembly projects, Geofront pipelines may act as a "custom GUI for any library" to an extent. In that sense, end users will only have to wait for a WebAssembly build, instead of a full application, before they are able to access a libraries content

Answers

The results show that this specific web-based VPL appears to be a feasible method for providing direct access to some native GIS libraries, and does offer a unique set of features not found in comparable visual languages. The significance of this method, compared to other web-based geometry VPLs, lies in the fact that it offers a lenient plugin system, in combination with a range of different GUI nodes, certain "dataflow VPL" properties, and a proposed zero-cost abstraction runtime. All of these features combined lead to a VPL which is able to directly connect GUI components with native GIS libraries, all while remaining scalable in principle. On a practical level, more work remains to proof this feasibility. The methodology developed by this study is only theoretically accessible and composable, based on achieved features. User-testing is required to confirm if this method indeed improves workflows, and actually saves time and energy of developers and end users. Moreover, the prototypical software implementation used is limited and not production ready. Both the fact that the 'no-boilerplate' plugin system cannot be used with C / C++ GIS libraries, and that backend execution is not possible yet, must be improved upon in future work.

Future Work

More Feature-complete VPL

- natively implemented, compiled to WebAssembly

Back

Thank you!

Questions?