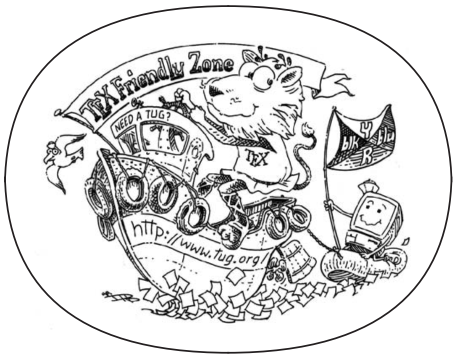
MASTER THESIS

EMIL MØLLER RASMUSSEN, IOANNIS ANGELIDIS AND DAVID NAGY



Creating a thin client based flood simulation tool based on open source software

June 2015 – version 1.0

Ohana means family.
Family means nobody gets left behind, or forgotten.
— Lilo & Stitch

Dedicated to the loving memory of Rudolf Miede.
1939 – 2005

ABSTRACT

Short summary of the contents. . .

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

Put your publications from the thesis here. The packages `multibib` or `bibtopic` etc. can be used to handle multiple different bibliographies in your document.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— **?** [**?**]

ACKNOWLEDGEMENTS

Put your acknowledgements here.

Many thanks to everybody who already sent me a postcard!

Regarding the typography and other help, many thanks go to Marco Kuhlmann, Philipp Lehman, Lothar Schlesier, Jim Young, Lorenzo Pantieri and Enrico Gregorio[1], Jörg Sommer, Joachim Köstler, Daniel Gottschlag, Denis Aydin, Paride Legovini, Steffen Prochnow, Nicolas Repp, Hinrich Harms, Roland Winkler, and the whole LaTeX-community for support, ideas and some great software.

Regarding LyX: The LyX port was intially done by *Nicholas Mariette* in March 2009 and continued by *Ivo Pletikosić* in 2011. Thank you very much for your work and the contributions to the original style.

1 Members of GuIT (Gruppo Italiano Utilizzatori di TeX e LaTeX)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

DRY	Don't Repeat Yourself
API	Application Programming Interface
UML	Unified Modeling Language

Part I

THE BORING STUFF

Introducing the project and it's various tidbits, have-beens and what-nots.

1 INTRODUCTION & PROBLEM STATEMENT

1.1 INTRODUCTION

Geographical Information Systems (GIS) have undergone an immense evolution during the last decade, evolving from an extremely niche product, to being used in a lot of contexts.

Where GIS used to be a niche product, more and more people have started using the capabilities of these systems. The interest in performing geographic analysis has sparked the development of open source projects software, capable of performing a wide variety of functions. Using open source technologies makes the cost of acquisition very low, enabling a whole new group of users access to these formerly expensive tools.

Of of the sectors traditionally involved with GIS is hydrology. Using geographic data to model hydrological phenomena has been done for many years, and it is very awesome by us and me. This type of modelling can be done either quick and imprecise, or slow and very precise, but all of these methods needs a fairly skilled technician to perform the analysis.

Why are we doing this analysis on floods?

1.2 PROBLEM STATEMENT

Providing a less technically capable user of creating

- Creation of a thin client based flood simulation and management tool using open source technologies

This problem should enable us to do a ton of work in absolutely no time, motherfucker!.

1.3 LICENSE

This project and all of it's items have been created as free and open source, and therefore follow the "GNU Software License"

2 METHODOLOGY

METHODOLOGY The works of this project officially started on the 2nd of February and the expected turn-in date was set on the 10th of June. This time-span provided the group with approximately four months to complete the development of this project and the report that accompanies it. As far as the working schedule is concerned, we decided as a group that a fixed set of days each week, where the group would meet and work on the project is best suitable. The rest of the days, each member of the group would work individually on pre-assigned tasks that would be discussed among the group in the next available meeting. Taking all the above into account, we set a weekly schedule that was followed throughout the completion of this project (table XXX). Monday Tuesday Wednesday Thursday Friday Saturday Sunday Meet – – Meet Meet – –

In an effort to maximize productivity and collaboration between group members, we decided to follow a classic software methodology which is called SCRUM. This choice was made due to the fact that this project includes significant amount of programming work and most of the members have experience working under this method. Being a sub-version of the Agile methodology, SCRUM basically adheres to a specific routine: Which tasks have been progressed since yesterday? Which tasks will be worked on tomorrow? Are there any obstacles preventing tasks from completion? Always keeping in mind this routine, three to four meeting were scheduled each week so that the group could discuss the direction and progress of the project. In addition, other means of connections (Dropbox, Google Drive) were established in order to maintain communication among the group members on the days that group meetings did not occur. During the meetings, group members could present propositions on how to enhance the project or seek assistance in the case where a task could not be completed. Developing the software and writing the report did not occur in parallel fashion, but extensive notes have been kept and a log was created in order to document the important issues and queries that occurred during the development phase. The main issue that occurred during the development phase of the project was that in some cases, code needed to be tested on the server to determine whether it performed without any errors. That fact, crippled the flexibility of the group on the occasions where we needed to test fixes for broken code. To be more specific, each time a fix was implemented, the server needed to be restarted in order to load the new script and test its new version. When restarting the server we had to make sure that no other group member was working on the server side, so a waiting gap existed between testing and fixing the code.

2.1 AGILE

2.2 TIMELINE

Functions

2.3 THE PHASES OF OUR PROJECT

Part II

THE GOOD STUFF

Introducing the various parts of theory used throughout the report.

3 THEORY

3.1 OPEN SOURCE GIS

The Open Source Definition is clearly set by the Open Source Initiative (OSI) in order to define which licenses can be defined as "Open Source". In addition to that, OSI provides certification to these licenses to indicate that they follow the open-source principles and comply with the Open Source definition. This definition states the following: The license should allow the sale or gifting of the software as a part of software distribution along with programs from several different sources. For such sale no royalty or monetary compensation should be required. ("The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale".) The source code of the software should be included in the software and it must be distributed freely along with the compiled form. In the case where the source code is not distributed with the software, an easily reached alternative must be provided, at most with a minimum reproduction cost. Modifications and derived works must be allowed and distributes as freely as the software itself. Modification of the source code can be restricted by the license only in the case that the license allows "patch files" distribution along with the source code for program modification. Software built from modified source code must be allowed to be distributed. Works derived from the source code may be required to have a different name or version number. Discrimination against persons or groups is not allowed All fields of endeavor must be allowed to use the software, unrestricted by its license. In the case where the program is redistributed, the same rights must apply without the execution of an additional license from those parties. In the case where the program is part of a specific software distribution, parties to whom the program is redistributed should have the same rights with those to whom the original program is distributed. Restrictions on other software, distributed with the licensed software must not be placed under restrictions. Access to the license must not be dependable on any individual technology pr interface The main reason open software exists is because the US Government, during the 1970ies and 1980ies implemented changes in the patent laws that allowed software and hardware companies to un-bundle software and hardware and the source code for software was under restricted access. For the reasons stated above the Free Software Foundation (FSF) was created (Grassmuck, 2004). Due to their widespread need, Geographic Information Systems software are also available to the public under the "Open Source" label. These software include a wide variety of open source examples that can be divided based on the functionalities the offer. The table below can provide with some insight on such categorization (table XXX).

Table 1: Categorization of GIS software. (Steiniger & Hunter, 2012) Some of the most common desktop GIS Software are the following (table XXX). Desktop GIS Software Developer GRASS GIS Neteler & Mitasova, 2008, Neteler, Bowman, Landa & Metz 2012 Quantum GIS Hugentobler, 2008 ILWIS / ILWIS Open Valenzuela, 1988, Hengl, Gruber & Shrestha, 2003 uDig Ramsey, 2006 SAGA Olaya 2004, Conrad 2007 OpenJUMP Steiniger & Michaud, 2009 MapWindow GIS Ames, Michaelis & Dunsford, 2007 gvSIG Anguix & Diaz, 2008 Table 2: Mature desktop GIS software (Steiniger & Hunter, 2012).

3.2 GRASS

Functions

3.3 HYDROLOGY IN GIS

Water rise

Choke point / pour point

Watershed

MCDA

3.4 PYTHON

Python is a high level low abstraction level programming language. Python emphasizes code readability. Python is compatible with all major operating systems, and usually comes pre-packaged with these. All python releases are open source. As a language, Python is highly extensible, which means that instead of coming prepackaged with all functionality built in, it has a certain amount built in, and expects the user to download / add necessary libraries as needed. The most commonly used versions of Python are 2.7 and 3.x. These differ for various reasons. The 2.7 release is a so-called legacy release, which means that it is not the worked-upon version, and wont see major updates. The 3.x is the newest version, and will be updated regularly. When working with ″older″ software, and libraries, it is most likely best to use version 2.7 as there will likely be compatibility issues when using a newer version of Python. Python has become a very popular programming language, and as such the possibility of using it with a variety of software packages has expanded greatly. Standard python syntax looks something like this: ” CODE SNIPPET ”

GRASS

GRASS functions can be used and manipulated by python scripting. Python can easily be used within the main GRASS shell, but it is also possible to create Python scripts that can call GRASS functionality from outside the main shell.

The functions and modules of GRASS, when used outside of an actual GRASS thingy, only work when a series of specific environment variables have been set.

GRASS session word should be used.

CODE SNIPPET

PyWPS

A Web Processing Service (WPS) is a standard defined by the Open Geospatial Consortium defining how inputs and outputs (also called requests and responses) for geospatial processing services should be standardized.

WPS Version 1.0 was released in June 2007, and WPS version version 2.0 was approved and released in January 2015.

The idea behind the standard is to standardize how inputs and outputs for geospatial processing services. It defines how a client can request the execution of a process, and how the output from the process is handled. Furthermore, it defines the interface that facilitates the publishing of geospatial processes and clientsâ discovery of and binding to those processes. The data required by the WPS can be delivered across a network or they can be available at the server.

In short, this should make it easier for people who want to publish custom geospatial calculations on the internet, using modern standards.

PyWPS connects the Web Browser, Desktop GIS, command line tools and working tool installed on the server. As working tool, GRASS GIS, GDAL, PROJ, R and other programs can be used.

PyWPS does not process the data by it self. PyWPS and GRASS can work together, but the setup has to initialize the above-mentioned variables throught some configurations set.

When requesting data from the server, the URL you send to the server, defines what kind of request you have made. The WPS enables a user to Describe a Process, Execute a Process and to Get Capabilities of the server, and the instances available. Similar to other OGC Web Services (such as WMS, WFS or WCS), WPS has three basic request types. Namely GetCapabilities, DescribeProcess and Execute.

Example strings for the three processes mentioned above:
http://webaddress/pywps/?service=WPS&request=GetCapablities
http://webaddress/pywps/?service=WPS&version=1.0.0&request=DescribeProcess&i
http://webaddress/pywps/?service=WPS&version=1.0.0&request=Execute&identifier

When an Execute request has been posted to the WPS, it will start processing on the server, and when it is done outputs will be provided encoded in XML.

The WPS standard actually requires the possibility for a user to keep track of how far along their process is, but as it is PyWPS isn't set up to this yet.

âCODE SNIPPETâ

Flask

Flask is a web application framework, written in Python and based on Werkzeug and the Jinja2 template engine. The framework is built around the idea of being as simple as possible. As such it only includes the bare-bone necessities, and expects the user to import third-party libraries that provide common functions.

Basically, this means that it is possible to create a dynamic web environment, by writing it in a combination of Python and HTML.

âCODE SNIPPETâ
MINIMAL APPLICATION

3.5 DIGITAL ELEVATION MODELS

3.6 WEB DEVELOPMENT

Web technologies

GNU GENERAL PUBLIC LICENSE: This program is free software; you can

Server side technologies

4 ANALYSIS

5 DISCUSSION

Part III

APPENDIX

A APPENDIX TEST

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

A.1 APPENDIX SECTION TEST

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

More dummy text

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse

LABITUR BONORUM PRI NO	QUE VISTA	HUMAN
fastidii ea ius	germano	demonstratea
suscipit instructior	titulo	personas
quaestio philosophia	facto	demonstrated

Table 1: Autem usu id.

Listing 1: A floating example

```
for i:=maxint to 0 do
begin
{ do nothing }
end;
```

viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

A.2 ANOTHER APPENDIX SECTION TEST

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

colophon

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LYX:

`http://code.google.com/p/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

DECLARATION

Put your declaration here.

Copenhagen, June 2015

Emil Møller Rasmussen,
Ioannis Angelidis and David
Nagy, April 29, 2015