# Community Mapping

Brian Michalski
Department of Computer Science & RCOS
Rensselaer Polytechnic Institute
Troy, NY 12180
michab3@cs.rpi.edu

## ABSTRACT

Community groups have many interests in generating highly localized maps of events, key locations, and other important markers as part of their respective missions. Sharing similar objectives, community groups stand to benefit from using and sharing similar mapping tools well suited to the model of free open source software. We discuss the development and deployment of a Community Mapping tool initially planned for use in community groups in Troy, New York.

## Categories and Subject Descriptors

J.1 [**Administrative Data Processing**]; K.3.4 [**Organizational Impacts**]

## General Terms

Management, Design, Human Factors.

## Keywords

open source, mapping, community involvement.

## 1.INTRODUCTION

The Community Mapping project was initially designed to help community groups in Troy, NY map locations of interest in support of a variety of different projects. Group members need a simple way to mark locations with additional metadata on a map, building overlays in a collaborate fashion. To accomplish this goal, we built a web application based on Flagship Geo [6], a Ruby on Rails geographic data framework. Current focus of the project is on extending the application to other community groups as a hosted cloud application, providing a free software as a service-style offering.

## 2.COMMUNITY COLLABORATION

Initial design discussions with a representative from a community group (conveniently, a professor at Rensselaer Polytechnic Institute) provided initial insight into the need for an application to quickly and easily build community maps for free. Local governments and affiliated organizations invest in software tools and integrated solutions to map data but many community groups, including those initially seeking this solution, do not have readily available access to commercial mapping products.

Recognizing the limited scope of an independent study, the Community Mapping project focused on providing a basic mapping solution that would quickly meet many of the needs of the local organization without requiring extensive training or time requirements from community volunteers. To accomplish this, the project was designed to be flexible, maximizing the use of general-use, optional fields. The software does not impose any validations or restrictions on what values can be used in the available fields except for the required latitude and longitude references.

After an initial draft of the software and user interface was available, community group representatives were given a brief demonstration and asked for feedback. The resulting conversations provided valuable clues into the different use cases the tool might have and also seemed to inspire additional projects that might be easier to tackle now that a simple mapping program was on the horizon. In addition to a discussion of the specifics of various projects that might benefit from the tool, the community members expressed a desire for several features like a need to print maps (an analog exchange this author regularly overlooks), and the ability to choose what maps are public or private. Continually iteration is underway to develop the identified features, at which point additional comments will be sought from the community groups.

Continual involvement of community members has been a key to developing a solution that would best serve their needs. By personally interacting with the community members planning to use the software, developers can gain extremely valuable insights into the problem the tool may be used to solve and better prioritize development efforts and focus on highly desired features.

## 3.APPLICATION ARCHITECTURE

Community Mapping provides each mapping project the ability to plot points on a map using many different layers. Layers are used as a logical grouping of points within each project, and each point typically represents a distinct location or unique event on the map. In many projects using the initial software, the number of layers used on a map is relatively static and small in number, while the points being marked change often.

### 3.1Layers

Each layer, unique to each project, is identified with a name, description, and graphical icon. These layers can be marked as hidden or visible on the project's main map or can also be viewed individually (e.g seeing all the markers assigned to the "public benches" layer). The layer icon is displayed on the map as the


*Conference'10*, Month 1–2, 2010, City, State, Country.

graphical marker for each contained point., providing an easy visual clue to associate marker that belong to the same layer.

### 3.2 Points

Points represent distinct units of information on the map and can contain a variety of metadata useful to the specific project. Points are located on a map with a latitude and longitude, which can be looked up by geocoding an address if available. Address geocoding is provided by the Google Maps API [2], and is executed via AJAX when a user is creating a point. Each point is represented with the icon inherited from being assigned to a specific layer. Points are required to have a name and layer assigned but optionally can have a description, datetime, and address used for geocoding. Further work may add the ability to dynamically create additional data fields for points within a project.

### 3.3 Projects

To support multiple independent maps on a single instance of the software, layers are points and separated into projects. Each project represents one map, with an independent collection of layers and points. Projects include reference to a specific geographic location through an assigned a latitude and longitude to automatically center the map and provide a default point of reference when creating new points. While projects by default are public, they can also be set as private so only authenticated users with access can view the map.

Each project can generate a static image of the map, suitable for use in presentations or printed material. In addition, projects can be exported as KML files [5] for offline display in software like Google Earth or for backup purposes. Future improvements may include the ability to import a KML file to a new project.

## 4. SOFTWARE AS A SERVICE OFFERING

As the initial application was being developed, it became clear through the use cases mentioned by the community groups that this application was not serving the unique needs of a single community. Paired with it's development as a Ruby on Rails web application, the Community Mapping project became an ideal candidate to be deployed and hosted in the cloud. Deployment in the cloud can greatly reduce the setup time and initial barrier of entry to organizations, especially those lacking technical expertise.

Given a budget of $0 and the lack of available production space for Ruby on Rails applications at Rensselaer Polytechnic Institute, the project has been hosted on Heroku through their free plan [3]. Static assets, such as the layer icons, are hosted on Amazon S3 [1] producing a monthly bill that can be paid in pocket change (e.g less than $1).

While the application enters a more public phase the cloud computing backend provides a great mechanism to scale if other community groups take interest in using the platform for local mapping. Significant use may increase the monthly cost to a noticeable amount, but that is a problem open source software solutions should like to have.

No specific components of the application require use of a cloud computing vendor to host the application. The dependencies are all publicly available open source libraries all of which can be setup locally on a Linux server or equivalent platform.

## 5. TECHNICAL CHALLENGES

The initial demonstration of the prototype tool got off to a rough start at the community group meeting. Wi-Fi access was not available, and the presentation of the web based application had to be carried out on the group's desktop computer running Microsoft Windows XP. During this demonstration in Internet Explorer 6 [4] several previously unseen bugs were exposed. Web application development, while much easier to maintain and standardize across platforms than traditional distribute and install desktop applications, is still not a completely standardized platform. Given the unknown resources of community groups, extra effort needs to be taken for the web-based application to be compatible with as many devices as possible, with particular attention paid to older hardware running outdated web browsers.

## 6. CONCLUSIONS

A final public release of the Community Mapping project is scheduled for December 2010 / January 2011. This initial public offering aims to adequately satisfy the needs of the local community groups that have participated in its development, but the application likely has uses far beyond that. Though a software as a service style offering, any community group can try out the tool to see what additional value it provides to their organization at no additional expense.

## 7. ACKNOWLEDGEMENT

The author would like to thank Sean O'Sullivan '85, RCOS, Dr. Mukkai S. Krishnamoorthy, Dr. Ken Rose, and the Troy Weed and Seed Program.

## 8. REFERENCES

[1] Amazon Simple Storage Service. Retrieved December 17, 2010, from Amazon Web Services: http://aws.amazon.com/s3/.

[2] Google Maps Geocoding. Retrieved December 17, 2010 from Google Maps JavaScript API V3: http://code.google.com/apis/maps/documentation/javascript/services.html#Geocoding

[3] Heroku Pricing. Retrieved December 18, 2010, from Heroku: http://heroku.com/pricing.

[4] Internet Explorer 6 Death. Retrieved December 18, 2010: http://www.ie6death.com/

[5] KML Standards, 2008. Retrieved December 10, 2010, from Open Geospatial Consortium: http://www.opengeospatial.org/standards/kml.

[6] MICHALSKI, Brian. Flagship Geo, 2010. Retrieved December 18, 2010 from Github: https://github.com/bamnet/flagship_geo

 