SDP 03 What’s Invasive!: A Model for Spatializing Data Sources

SPD 03.1 Overview
“What’s Invasive!” can be viewed more broadly as test case of the OurPixel ideals put into motion. OurPixel’s goals are to promote citizen science via easy-to-use data sources and tools that traditionally have not been easily accessible to the general public. “What’s Invasive!” is focused on leveraging geospatial tools and smart-phone capabilities to help communities anywhere help locate invasive plant species. By making geo-tagged observations through these devices, users can not only track them via the web, but also interact with the data. For this reason, it was necessary to build a standard set of geospatial and visualization tools that could be used in any number of projects.

This discussion will outline the design of a highly flexible open-source web-based mapping system that is to meet the data demands of citizen-science campaigns. With the recent developments in open-source software for geospatial data, we have developed a system that is easily expandable, enables users to access GIS data layers and interact with submitted data.

SPD 03.2 Approach
The breadth of a system that embodies OurPixel required considerable input from team members. Several rounds of proposals went out in order to encapsulate an open review and acquire group input. It was important that our choices were dictated by ease-of-use, compatibility between various software components, and tied together by a unifying framework. Some of the requirements for such a system are as follows.

We want tools that are open-source, allow for rapid development, and interoperability. These should not be unique to “What’s Invasive!”, so what is built is easily replicated for other projects. This system should be an expandable framework in a way that accommodates changes in a simple, efficient way.

Because of the geospatial nature of “What’s Invasive!” (and more broadly to OurPixel), it is desirable to have intuitive and unrestrictive geospatial tools that allow for easy access to data layers such as shapefiles, raster data, and KML. It is important that users can create and store data, tag points, draw polygons, upload photos, through an effective user-interface. For data storage, we need a database that should be “spatial”, so that we can take full advantage of the location-based nature of the data.

Social-networking features are highly desirable and should reflect the goals of the OurPixel project. This includes an ability to create a community around the project. An implementation might include tools such as blogs, photo management, groups, and discussion boards.

In terms of system design, Figure 1, these features fall into 4 categories, namely,

- Server-side software and Databases
- Middleware and Web Frameworks
- Geo-spatial tools and plugins
- User-Interface

SPD 03.3 System Description

Back-end and Database
The back-end of the system is built on top of a Ubuntu 9.10 Linux-Apache-M. This allows for easy access to our required open-source software and is a very stable platform. Data is stored through the use of PostgreSQL and PostGIS. PostGIS provides an interface for PostgreSQL to spatialize our data. It is highly interoperable with our other choices, in order to accommodate a natural flow between components.

Web Framework
The key component that connects all of our tools is Django. It is a high-level Python web framework that embodies rapid-development, functionality between components, and is open-source. It utilizes functional and inter-operable
applications to speed up the development process, such as GIS and Social 2.0 features. It is run on an Apache server using the Web-Service Gateway Interface.

**Geospatial tools**
The GIS branch of Django, GeoDjango, provides a geographic web-framework for building GIS web-applications. It utilizes powerful geospatial libraries (GEOS, PROJ, GDAL) that work together with all levels of the system. GeoServer gives us a natural tool to serve up more complex data.

**User Interface**
The OpenLayers Javascript library offers extensive support, extendibility and interoperability between our components (GeoDjango, PostGIS). It allows us to access any number of map tile servers such as Google Maps, Yahoo! Maps, and OpenStreetMaps. It has strong functionality that connects to all of our potential data sources.

Figure 2 gives an overview of how these pieces interact with each other.

### SPD 03.4 Accomplishments
The core framework for an expandable geospatial-social system is complete. By using the Django web framework, we have built a core foundation that is easily modified and replicated for other campaigns. It is the heart of the system and connects to all other pieces in a simple, yet powerful way. It connects easily to our robust geospatial tools and opens up many avenues of future development. A map-server in the form of GeoServer is available to serve up shapefiles, KML, and raster data. Our user-interface utilizes OpenLayers as an effective way of connecting the user and data available. However, there are other components that still need further development; details for improvements on the current system are presented below.

### SPD 03.5 Future Directions
There are several areas of development that would need to be implemented in order to fully expose the capabilities of the system and OurPixel's broader goals.

**Social Networking Features**
Due to time constraints, the social networking and community aspects were not implemented. This could be accomplished without many delays, because of the nature of the Django framework and Pinax. Pinax is an application plug-in for Django that offers a plethora of features to "socialize" any project.

**Mapping-Geospatial Features**
The current map interface through OpenLayers/Google Maps could be improved through the use of JavaScript frameworks, ExtJS or jQuery. Additionally, more data layers and a more intuitive interface to access these layers would be ideal.

**Data Feeds**
There should be better communication or integration between the current setup of databases and data sources. There is data in multiple locations, e.g., Twitter and Flickr, and connecting them would be parsimonious. This could be done via a reusable application with Django.

**Analysis and Visualization tools**
There is a need for advanced analysis and visualization tools for "What’s Invasive!". These should go beyond basic charting and help user’s explore the data in an intuitive and statistical way. The use of R via Python, in conjunction with Django, should provide a direct path for this.