Plant Disease System Design (Prototype)

Max Guo
Email: maxlovesbbq@gmail.com

Background:

A plant disease is usually defined as abnormal growth and/or dysfunction of a plant. Diseases are the result of some disturbance in the normal life process of the plant. Diseases may be the result of living and/or non-living causes. Biotic diseases are caused by living organisms (e.g., fungi, bacteria, and viruses). Manage and predict plant disease has long been a challenge for farmers.

AAFC(Agriculture and Agri-food Canada) is keen to develop a live Web-GIS system to help indicate potential risk for certain areas based on a spatial model.

1.Planning

The idea is to let user select an area of Interest first, along with some other inputs the system will generate either a field based report or a dataset representing the plant disease risk.
The key for this system are: first understanding the data inputs needed for the model; second find a good system model to integrate all the data and service together.

2.Analysis

2.1 Data Gathering
FInding the required data for the model is important, for this model the data we need are: History of Disease, Soil Moisture, Crop Density, Rainfall, Temperature, Crop phenology, Cropping History, Micro-topography, Soil Texture and Soil PH.

Risk Factor	**About**	**Data Source**
History of Disease	Disease incidence in last host crop	User Input
Soil Moisture	Number of days soil is at or near field capacity	Environment Canada
Crop Density	Density of the vegetation	User Input
Rainfall	At least 5-10 mm on more than 2 days per week	Environment Canada

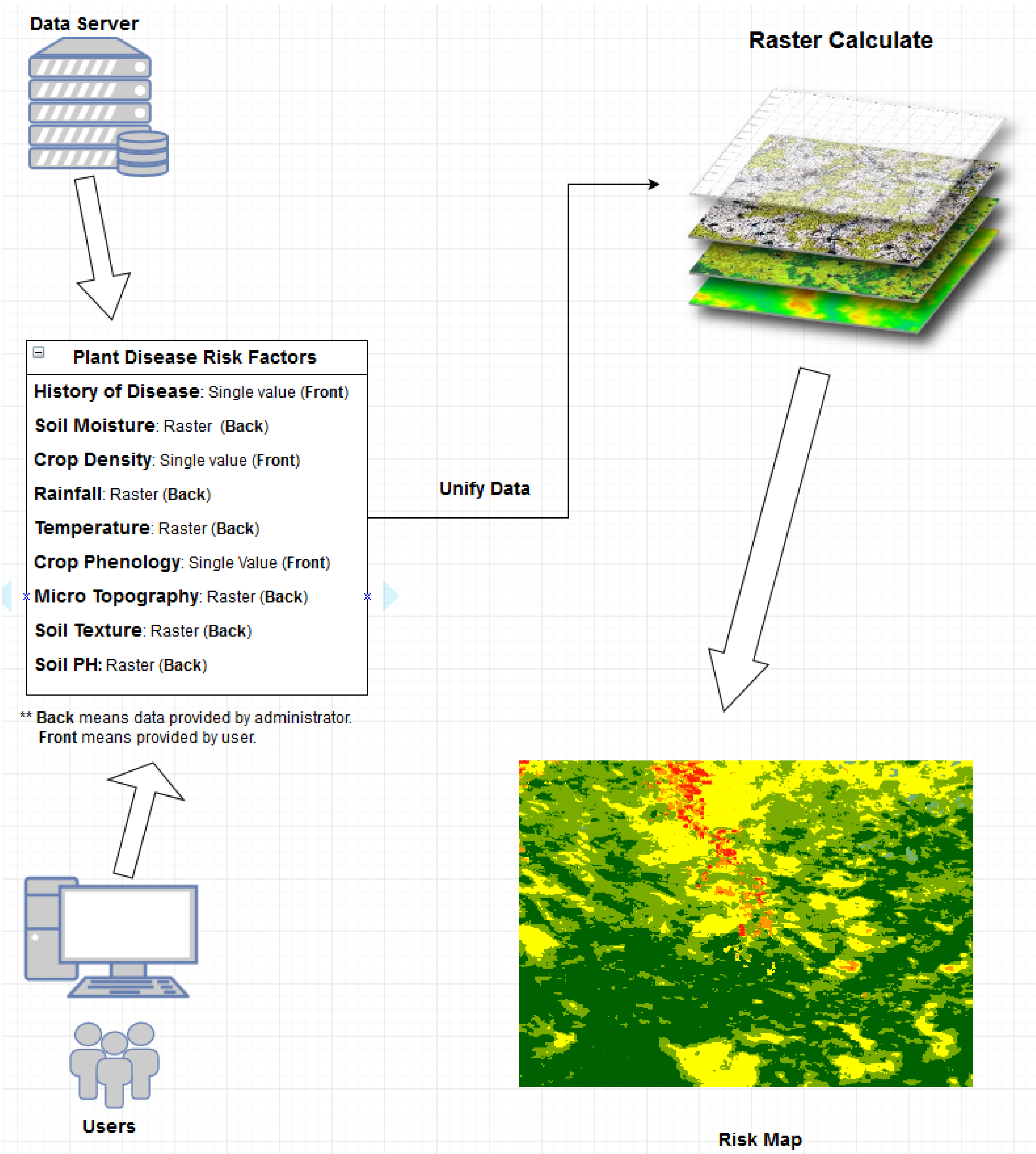
Temperature	Min temp > 10oC and Max temp < 30oC	Environment Canada
Crop phenology	Date of flowering	User Input
Cropping History	Number of years since last canola crop	Agriculture and Agri-food Canada
Micro-topography	Elevation	Agriculture and Agri-food Canada
Soil Texture	Nominal (Clay - Loam)	Agriculture and Agri-food Canada
Soil PH	Numeric (PH < 6.5>	Agriculture and Agri-food Canada

2.2 Data Fusion

From the Table above, we have a clear understanding of what data are going to be needed for the plant disease model. The data comes from the “User input” would be either a single value or a condition indicator and the data comes from “Agriculture Canada” and “Environment Canada” are going to be raster datasets with different cell resolution, other than this, the area of interest need also be provided by the user. In order to put data in use, the next thing is to consider how to integrate and unify these data.

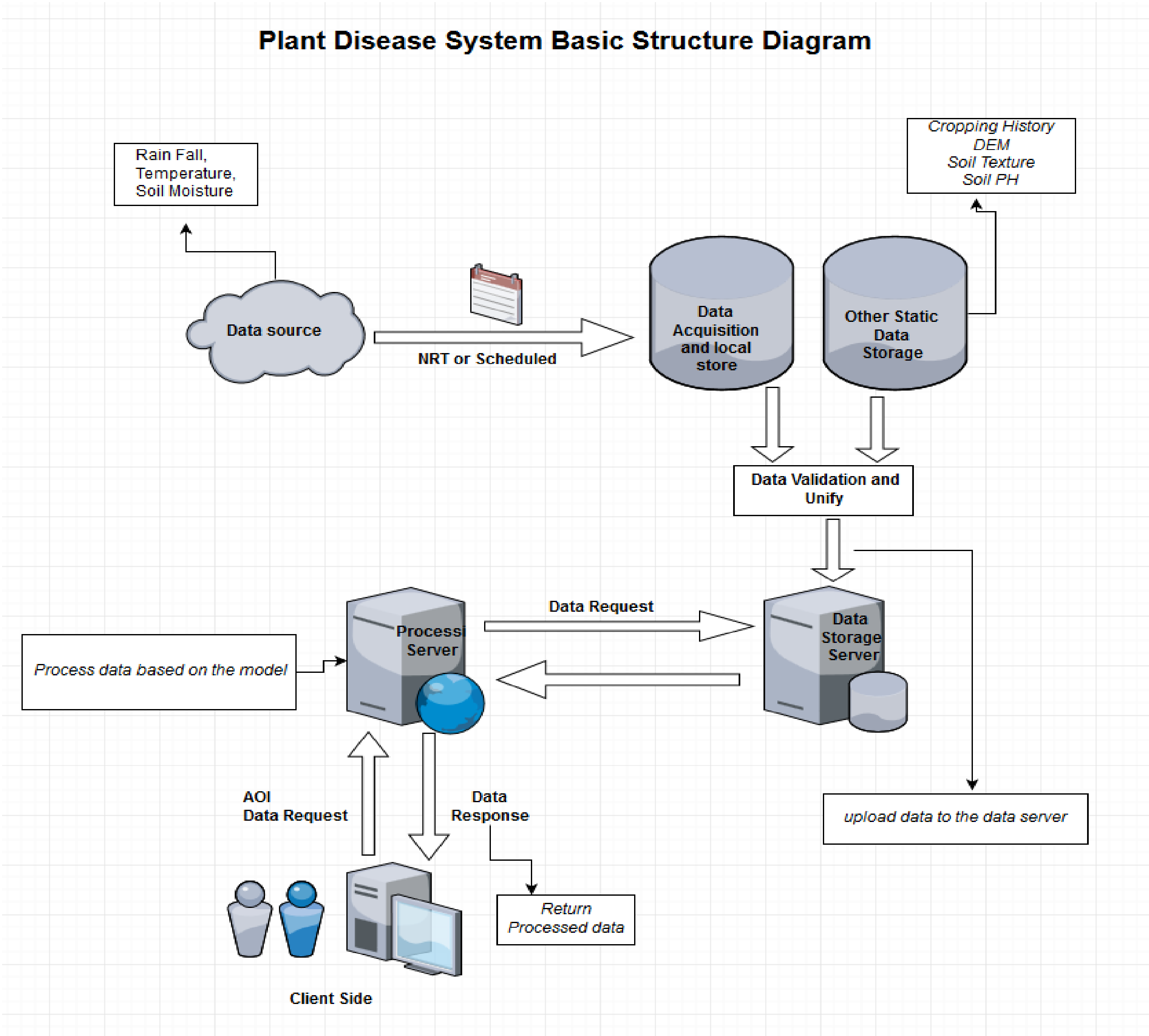
All the data needs to be converted into the same format, raster calculation is widely used for geospatial analysis, we are going to to convert all of our data into raster format with same dimensions and same resolution. After, do a raster calculation for the layers and generate a risk indication result. The user side data will be transferred into a single value raster band and the all raster files from different resource will need to be resampled and reprojected under same standard.

Following is an illustration showing the idea of the process.


Data Server
Raster Calculate
Plant Disease Risk Factors
History of Disease: Single value (Front)
Soil Moisture: Raster (Back)
Crop Density: Single value (Front)
Rainfall: Raster (Back)
Temperature: Raster (Back)
Crop Phenology: Single Value (Front)
Micro Topography: Raster (Back)
Soil Texture: Raster (Back)
Soil PH: Raster (Back)
Unify Data
** Back means data provided by administrator.
Front means provided by user.
Users
Risk Map

3. Design

For the system we decided to use LAMP model. There are three parts of the system: The client side, the processing server and the data storage server. Client side is the front-end GUI, where user input data and make requests. Large static raster data is stored in the data storage server and processing is done on the processing server using the data come from both data storage server and user side. Data is requested from data storage server via URL request, then made available for download on the client side and process on the processing server. Visualization of this data can be offered as well, but the focus is on processing and data download. A Diagram below is made to help illustrate the idea of the system.



4. Implementation

4.1 Data storage server

The data storage server is going to be the geoserver. GeoServer is a Java-based software server that allows users to view and edit geospatial data. Using open standards set forth by the OGC, GeoServer allows for great flexibility in map creation and data sharing.

4.2 Client side

The basic map display will be based on Openlayer framework. By using openstreetmap layers, client will be able to see and interact with the storage server. The spatial data is transferred using WCS (web coverage service), a digital geospatial information representing space/time- varying phenomena; the WCS could be converted into raster format datasets.

The front-end GUI is programmed using HTML, CSS and Javascript, the data received from the front-end will be encoded into a data stream and pass to the data processing server.

4.3 Data processing server

There are two parts of data processing server: data requesting from data storage server and actual data processing.

Data requesting is done in PHP, by sending a WCS http request to the Geoserver, the processing server will be able to get the raster data and store it locally, in this model, the data would be cropping history, DEM, Soil texture and Soil PH.

Data Processing is done in Python, using OSGO GDAL library. The data is transporting between process as a JSON Encoded data stream.