Nicole C. Weber
10 July 2008

Software Requirements Specifications

a) *Functionality.*
This software provides a method for on-site responders to upload images and information from UAVs at a disaster site, and allow remote access of that information as well as modification capabilities. The expected users are UAV operators and other on-site emergency responders. Input from Sam Stover (FEMA) regarding field workers' needs has been taken into account. Stover's requirements included: the ability to insert text and other pictures from the user's personal cameras, the inclusion of GPS coordinates and time & date stamps for forensics, and the ability to go back and augment these entries after leaving the site. The functions are as follows:

1. **Initialize Application**: Upon initializing the application, the user will be able to select which site's data to view (for example: "Berkman Plaza II Parking Garage" or "Hurricane Katrina"). Upon selection, the web application will localize to that location and load the data associated with that site.

2. **Create New Site**: User is prompted for the following information:
 a) Site name
 b) Approximate central GPS location for window-viewing purposes
 c) Additional description of activity and events at site

The data are requested via JavaScript and passed via PHP to a new folder containing the information in KML format. Once a site is created user can create and view its Placemarks.

3. **Create Placemark**: User is prompted for the following information:
 a) Image (jpg file type)
 b) File name of image (String type)
 c) GPS data (numeric type: latitude, longitude, altitude, date and time)
 d) Compass data (numeric type: heading, pitch, roll)
 e) Description of feature(s) noticed at site but not visible in image (String type)

The data are requested via JavaScript and passed via PHP to a script that converts them to a KMZ file. The KMZ file is located in a folder of features that are automatically opened when the application is initialized.

3. **View Placemark**: After Placemark is created, user can click on it to view all inputted information in a pre-set format. There will be an option in the Placemark to edit its description.

4. **Edit Placemark**: While viewing Placemark, there is a button to add information to the description. User is prompted for additional description and can insert text and photos using HTML. This is passed through PHP and added into the existing KMZ file containing the placemark.

b) *External interfaces.*
As per the advice of Stover, the software must be Google-compatible. This software is an extension of the Google Earth Browser Plug-in software. Using the Google Earth API, JavaScript and PHP, the software is web-embedded and allows the user to input the following relevant information (as specified above). Not all of the information is required as it is not all expected to be readily available (for example, GPS data are not currently obtainable from the UAV images). Once the information is received it will be converted to a KMZ file, a zipped file consisting of the image and the KML code defining its placemark on the Earth. Users can click on the Placemark to view all information contained in it. Additionally in the Placemark there will be a function to

edit the description field, allowing the user to input additional pictures (externally hosted?) and information.

c) *Performance.*
The speed of the software will be dependent upon the user's internet connection. It should take a minute or less for the web application to load. Depending on the properties of the computer and the server hosting the data, the application may take longer than a minute to load. After submitting information, it should take a minute or less for the Placemark to be created or edited.

d) *Attributes.*
Building upon the Google Earth Browser Plug-in requires less memory use than running the Google Earth application. It additionally requires less application development time since Google Earth Browser Plug-in has an API for interactivity. Additionally using the Browser Plug-in allows remote access from computers not pre-loaded with the application. Downloading the Browser Plug-in is still necessary but uses less memory than downloading the Google Earth application.
For security purposes, the web application should be hosted in a directory that is username- and password-protected, to prevent unauthorized individuals from inserting incorrect or inaccurate information.
To allow external individuals to access the information, a smaller version of the application, without the Create Placemark/Edit Placemark functions, can be hosted in a public directory.

e) *Design constraints imposed on an implementation.*
Google Earth Browser Plug-in requires that a specified JavaScript API be used.
The Google Earth Browser Plug-in currently only works with these browsers in Microsoft Windows XP and Vista: IE 6.0+, IE 7.0+, Firefox 3.0x, 2.x or 2.0x, Netscape 7.1+, Mozilla 1.4+, and Flock 1.0+ (see http://code.google.com/apis/earth/documentation/index.html) but will be extended to other operating systems in the near future.

f) *Timeline.*
July 16: Learn and gain proficiency in the Earth/JavaScript API
July 18: Create application with Initialize and View Placemarks functions
July 25: Implement Create Placemark function
July 29: Implement Edit Placemark function