GeoVTag

Michel Deriaz

University of Geneva, Switzerland
{firstname.lastname} [at] unige.ch

Abstract. This paper presents GeoVTag, an application running on a mobile phone that allows the user to publish anywhere on Earth virtual tags. Every user in the neighborhood of such a publication point will get the message. It is a kind of blog, in which editors and readers share the same physical place. GeoVTag allows tag edition and reading, but has also some additional functionalities like a digital compass, a radar to find and display graphically all neighboring tags, a tracker to record GPS data, a marker function that sends automatically time-to-time a tag, and a "goto" function that directs a user to a specific tag by showing graphically the direction to follow and the remaining distance.

1 Introduction

Spatial messaging, also called digital graffiti, air graffiti, or splash messaging, allows a user to publish a geo-referenced note so that any other user that attends the same place can get the message. For example, let us consider the community of the Mt-Blanc mountain guides. The members would like to inform their colleagues about dangers in specific places or about vacancies in refuges. One guide can publish a geo-referenced message that informs about a high risk of avalanches, and any other guide that attends the same place will get the warning, and comment it if necessary. Spatial messaging is a kind of blog, in which editors and readers share the same physical place.

A paper that gives a general description of this concept and that presents different hypothetical scenarios is given in [1]. It describes also a trust and security architecture which aims to append to each tag some trust information, so that a reader knows how reliable a tag is. GeoVTag is a first move to its implementation. It will serve as a test platform to evaluate the future trust models. Currently it allows tag edition and reading, as well as some additional functionalities like a digital compass, a radar to find and display graphically all neighboring tags, a tracker to record GPS data, a marker function that sends automatically time-to-time a tag, and a "goto" function that directs a user to a specific tag by showing graphically the direction to follow and the remaining distance.

Technically, posting virtual tags is quite simple. During a tag edition, the application gets also the GPS position and adds it as a meta-data. The whole is then

sent via Internet (using the HTTP protocol) to a tag server. For the tag reading, the principle is similar. The mobile user sends its current position to the server (still using the HTTP protocol) which returns all the available tags at the given position and its neighborhood. The size of the neighborhood is specified during the request.

We implemented this application using the java micro-edition language. This guarantees us more independence towards the running device. For the tests, we started with the Motorola A1000 smart phone, but where constrained to abandon it. Although it owns an internal GPS, the access to the java location API was restricted to digitally signed applications. The second main limitation of this device is that it follows only the CLDC 1.0 specification (instead of CLDC 1.1) and has therefore no floating point operations, which is a considerable drawback for our kind of application. We chose therefore to couple the Sony-Ericsson K750 mobile phone with the GlobalSat BT-338 Bluetooth GPS for our tests.

This paper is structured as follow: Section 2 gives some information about the GPS device, section 3 indicates how to install GeoVTag, section 4 is a direction of use, section 5 presents three other tools that have been developed around this application, section 6 talks about the evaluation of this software, and section 7 concludes by indicating how this software will be used as a test platform for future work.

2   GPS device

Using a Bluetooth GPS brings the advantage that our application can be used on a wide range of different devices. A Bluetooth GPS usually sends its data in NMEA sentences. Our application deals with two of these sentences, GGA and VTG. Typically a GGA sentence looks like this:

```
$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47
```

where 123519 is the time in UTC, 4807.038 is the latitude, and 01131.000 is the longitude. The reader that is unfamiliar with NMEA sentences can get more information at [2]. Our GPS has typically a precision of about 10 meters. It uses the Sirf III technology and therefore only needs -159dB as signal strength, against -175dB for a Sirf II GPS. Concretely it means that such a GPS works also in difficult situations, like behind an athermanous windscreen or in small roads surrounded by high buildings.

3   Installation of GeoVTag

Tests have been made on the Sony-Ericsson K750 mobile phone and the GlobalSat BT-338 Bluetooth GPS, but all the devices that have the following characteristics should work:

- A mobile device with a J2ME (Java 2 Micro Edition) interpreter, supporting MIDP 2.0, CLDC 1.1, JSR 135, JSR 75, JSR 82 and having a screen of minimum 176x220. Specifications about any phone can be found at [3].
- A Bluetooth GPS sending NMEA messages. GGA sentences are required, VTG sentences are optional and only required for specific functions. Information about NMEA, GGA and VTG can be found at [2].

To test the software, it is possible to download all the required files from [4]. Note that this paper describes version 0.2, so it is possible to have little differences with the current version, which is frequently updated. The application comes as a .jar file and its corresponding .jad file. By default, a user-defined property called "url" contains "http://127.0.0.1:8080/geovtag/" as value, meaning that the remote tag server runs on the same machine. If your IDE does not support the modification of this value (in the .jad and the .jar), proceed as follow: Modify manually the value in the .jad file, unzip the .jar file, modify manually the value in the META-INF/MANIFEST.MF file, and finally re-zip the all.

The installation itself is then completely device-dependant. If you want to test the application on the Sun emulator [5], you can simply use the "File - Create project from JAD/JAR" command, and then tick off the "CLDC 1.1" as well as "Bluetooth/OBEX" checkboxes in the settings.

4   Using GeoVTag

4.1   General information

This part gives some general information that is directly related to the application. The starting point of the application is a main menu (see Figure 1) form which all the other functions are accessible. As a GPS sends the same sentence type (GGA, VTG...) maximum once in a second, the updates are also done at this rate. Some functions display information in blue if it is fresh or in red if the last data obtained from the GPS is older than 3 seconds.

Because this application is not digitally signed, you will be prompted a few times when you launch the program, asking your authorization for accessing the Internet, the Bluetooth GPS or your local file system. Just answer "yes" to these questions. You them arrive in the main menu of the application (see Figure 1). Use the navigation keys to select a function.

4.2   Tag function

This function shows you the current time (from GPS), your current position and all the available tags at the current position (See Figure 2). The current position is actually a circle with a 20-meter radius centered on the point given by the GPS.

The menu provides 3 commands:

- Tag edition: This screen is used to create a new tag. Note that when this screen is displayed, the current location is frozen. It means that even if you move between the time you start editing your tag and the time you send it, the tag will be posted at the location you were when you started editing. You can yet update your current position with the "Update GPS data" command or enter it manually with the "Select coord." command. The "Send tag" command sends the tag to the server, and the "Main menu" command brings you back to the main menu.
- Save tags: If you use a local tag server (see 4.6 - "Options function"), the tags are not saved automatically (otherwise you would be prompted each time by the system in order to give write access to the local file system). This command saves all the new tags in the file specified in the options.
- Main menu: Brings you back to the main menu.
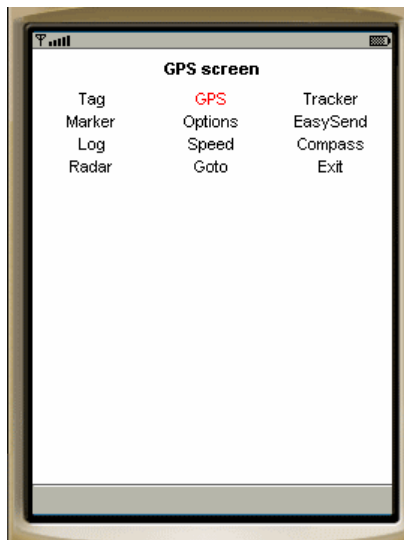
Figure 1	Figure 2

4.3 GPS function

Displays information given by the GPS (See Figure 3). From the GGA sentences we get the time, the position, the altitude (in meters), the quality (0 = no fix, 1 = GPS fix, 2 = differential GPS fix), and the number of satellites used to get the fix (minimum 3). If the GPS sends also VTG sentences, the display adds the current heading (N = 0°, E = 90°, S = 180°, W = 270°) and the current speed (in km/h).

The menu provides two commands:
- Connect GPS: This command launches a search for about 10 seconds of all neighboring Bluetooth devices. Use the "Back" menu to cancel the operation, or the arrow keys to select a Bluetooth GPS from the returned list. The URL of the

Bluetooth GPS is then saved so that the application tries to reconnect directly the next time it starts.
- Main menu: Brings you back to the main menu.
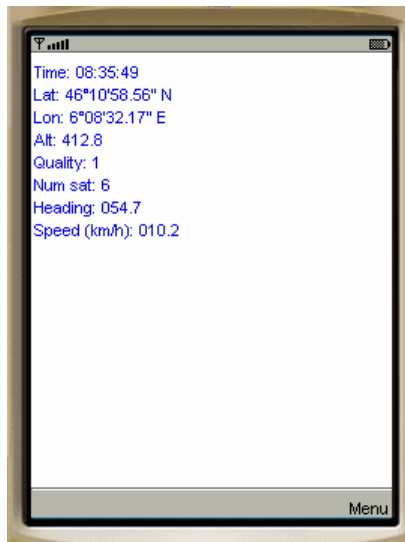
Figure 3

Figure 4

4.4 Tracker function

Records in a file the GPS data every few seconds (see Figure 4).
- State: Indicates the state of the process. Can be "Not started yet", "Running", "Stopping" or "Stopped".
- Session counter: Counts the number of positions that have been recorded since the last time the data have been saved.
- Global counter: Counts the number of positions that have been recorded since the last time the application has started.
- File: The file that will contain the data. Click on it to browse the local file system. See also 4.14 - Browsing the local file system.
- Interval: Specifies how many seconds the application pauses between two records. Note that the time of the recording itself is not counted in this time, meaning that the real interval between two records is actually the number specified here plus the time your device needs to make a record.

Commands:
- Start: Starts the recording.
- Stop: Stops the recording.
- Record now: Records now the current data. Can be used while the process is running or not.
- Save: Saves the recorded data in the specified file, and clears the session counter.

- Main menu: Brings you back to the main menu. The process won't be stopped by this command.

4.5 Marker function

Sends automatically a tag every few seconds (see Figure 5).
- State: Indicates the state of the process. Can be "Not started yet", "Running", "Stopping" or "Stopped".
- Content: Content of the tag. If the content ends with a hyphen followed by a number, this number is automatically incremented. For example, if the content is "Test-00" then the tags will be "Test-00", "Test-01", "Test-02"... With "Other-0023" we will have "Other-0023", "Other-0024", "Other-0025"...
- Interval: Specifies how many seconds the application pauses between two records. Note that the time of the recording itself is not counted in this time, meaning that the real interval between two records is actually the number specified here plus the time your device needs to make a record.

Commands:
- Start: Starts the process.
- Stop: Stops the process.
- Send now: Sends a tag now. Can be used while the process is running or not.
- Main menu: Brings you back to the main menu. The process won't be stopped by this command.

4.6 Options function

Specifies the options for the application (see Figure 6).
- Time diff.: The number of hours to add (or subtract if negative) to the time given by the GPS (which is UTC), during display. Note that this affects only how the time is displayed on the screen, not the time that will be taken into account when creating a tag or making a track.
- Position format: Similar than "Time diff". "°" represent the data in degrees, "° ' "" represents it in degrees, minutes and seconds.
- Tag server: Specifies if the tag server is local or remote. If local, then the tags are not sent over the Internet (there are therefore no costs) but recorded locally in the file specified in the "Local tag file" field. If remote, then the tags are sent via Internet to the remote tag server.
- Local tag file: If the tag server is local, then the tags are recorded in this file. Note that tags are not automatically saved in this file, but they are saved only by the "Save tags" command (in the Tags function). This file is read when the application starts. When the user changes the tag file, then all the tags in memory are erased (including the tags that the user didn't saved!) and the new tags are loaded in memory. See also 4.14 - "Browsing the local file system".
- Flash delay (sec): Many mobile phones put their screen in stand-by after a while if there is no user input. To keep the screen bright, specify here how many seconds to wait before the application flashes the screen. Note that some mobile

phones loose the connection with the GPS when they go in a stand-by position; this option can be set in order to avoid that.

Commands:

- OK: Confirms the modification and shows the main menu.
- Cancel: Cancels all the modifications and shows the main menu.
- Default values: Sets the default values.
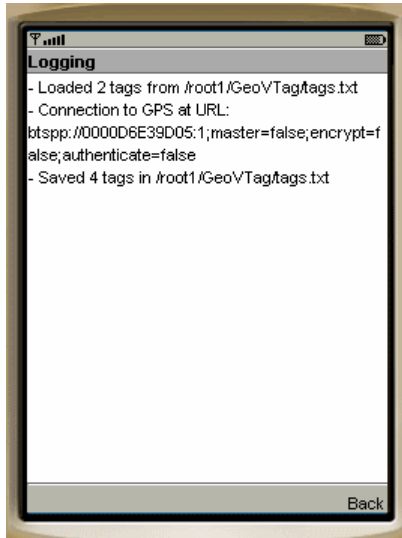

Figure 5

Figure 6

4.7   EasySend function

Sends a tag by just pressing a single key. The user links each numerical key (0 - 9) to the text he wants to send each time he presses that key. It seems important to remind here that, if you use a local tag file, then the tags are not automatically saved.

4.8   Logging function

Logs some events, like tags loading or saving (see Figure 7). This function is mainly used to make sure that tags have been correctly loaded or saved.

Commands:

- Back: Shows the main menu.

Figure 7

Figure 8

4.9   Speedometer function

Indicates the current speed (see Figure 8). The speed information is extracted from VTG sentences. This value is computed by comparing two succeeding positions. Because of the precision limitations of the device, it is possible to print an average of the last values. The printed value is then more precise but reacts slower to speed changes. In addition to the speed average, this function prints also the current speed (as gotten by the GPS device) and the number of values taken for the average computation.

Commands:

- Back: Shows the main menu.
- Average: Shows a list allowing the user to precise the number of values to take for the average.

4.10   Compass function

Shows the direction of the North Pole (see Figure 9). In digital compasses, the direction is computed according to the current heading of the user. Like for the speedometer, the GPS device must be able to send VTG sentences. To find the North Pole, walk straight forward for a few seconds while holding your mobile device in front of you. The red line on the screen indicates the North Pole. Note that it indicates the true North, and not the magnetic North, as traditional compasses do. This value is computed by comparing two succeeding positions. Because of the precision limitations of the device, it is possible to print an average of the last values. The printed value is then more precise but reacts slower to direction changes. In addition

to the direction of the North Pole, this function prints also the current heading (as gotten by the GPS device) and the number of values taken for the average computation.

Commands:
- Back: Shows the main menu.
- Average: Shows a list allowing the user to precise the number of values to take for the average.
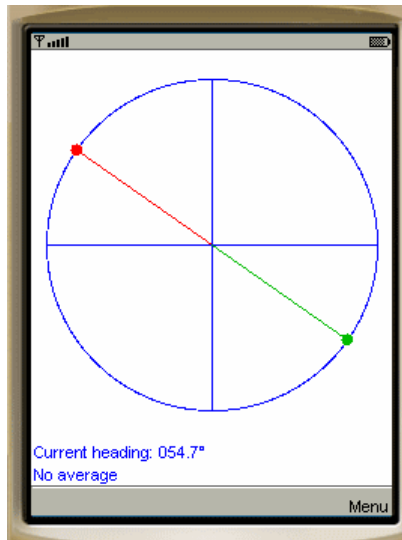
Figure 9
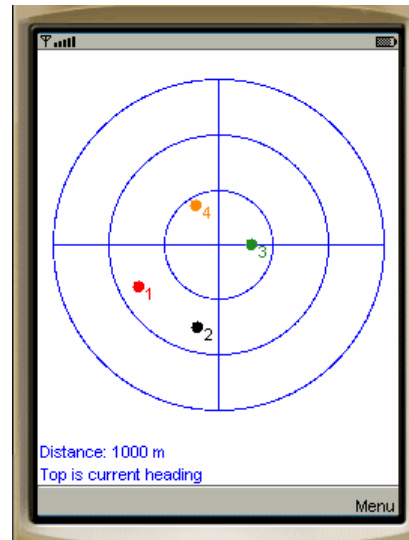
Figure 10

4.11 Radar function

The radar function shows graphically the positions of the neighboring tags (see Figure 10). The distance printed at the bottom of the screen indicates the number of meters between two circles. The second line at the bottom indicates whether the top of the screen corresponds to the North Pole or your current heading. In the second case, the current heading is given by the GPS if it understands VTG sentences. If you do not move fast enough, the printed dots, whose positions are computed according to your current heading, become leaping. It is therefore recommended to choose the North Pole option and use the compass function to find more precisely where the North Pole is.

Each time a new tag enters in the covered area (three times the value specified in "Distance"), it gets the smallest available number. This number determines its color, and allows the user to recognize it in the "details" screen. When a tag leaves the covered area, its number is made available for the next tag.

Commands:
- Details: Shows detailed information about the visible tags.

- Params: Allows the user to specify some parameters like the distance between two circles, whether the top of the screen represents the North Pole or the current heading, and whether numbers should be shown. If the alarm checkboxes are ticked off, then a sound (or/and a vibration) is played each time a new tag appears on the screen.
- Freeze GPS: Data that comes from the GPS device are not updated anymore.
- Unfreeze GPS: Cancels the "Freeze GPS" command.
- Main menu: Shows the main menu.

4.12 Goto function

This function allows you to find a specific tag, given its name or a part of its name (see Figure 11). When you activate this function, you are prompted with a form asking you the name of the tag (or a part of its name) and the radius of search. The "Search" command looks then for all the available tags that contains the specified pattern (case insensitive) and which are not further than the specified distance. The result is presented in a list. Select one item. The program shows then graphically and continuously the direction to follow to reach the tag, and precisely at which distance it is. The first line of text indicates this distance between you and the tag, and the second line indicates whether the top of the screen is the North Pole or your current heading (see 4.11 - Radar function).
   Commands:
- Details: Gives detailed information about the tracked tag.
- Params: Allows you to define whether the top of the screen represents the North Pole or your current heading.
- Back: Returns to the list resulted by the last search.
- Close: Closes this function and goes back to the main menu.

4.13 Exit function

The exit function closes open connections, stops running operations, and finally exits the application. It is important to remind here that local tags are not saved automatically when the application exits.

4.14 Browsing the local file system

The application includes a file browser that allows you to create new files or directories, delete them, and to select a file. It is used to select the local tag file in the options or to select where to save data from a track. The title of the screen indicates the current directory (see Figure 12). "d .." means the parent directory. Actually, all the lines starting with "d" (and ending with "/") are directories, while line starting with "f" are files and lines starting with "r" are roots. Typically, a mobile phone has two roots, the first is the build-in memory of the phone and the second is an additional

memory stick. We recommend saving tag files and track files in the memory stick rather than in the phone memory.
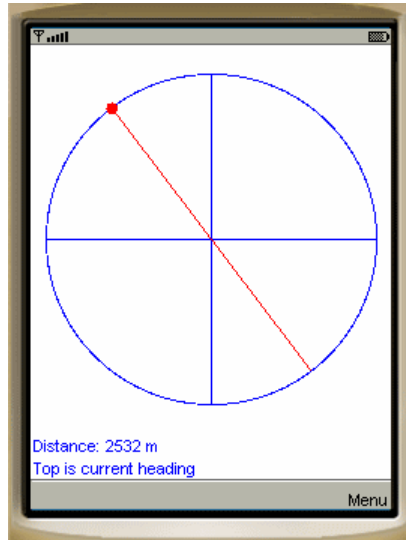
Figure 11
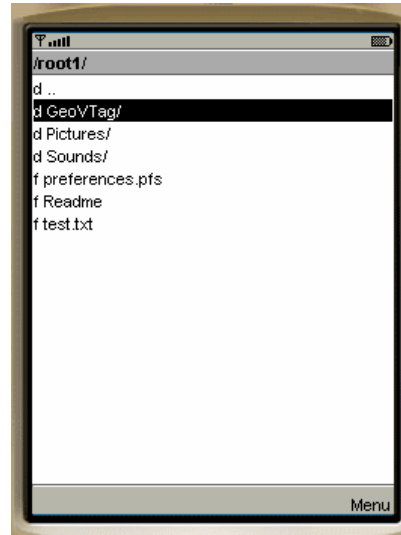
Figure 12

4.15 Saving parameters

The values entered in the Option screen, as well as the ones entered in different fields (mostly in parameters screens) are saved automatically in a MIDlet record store. You will not be prompted for an authorization like it is the case each time you access the local file system. Record stores stays even if the application is reinstalled. To delete the record store associated to GeoVTag, delete the application before reinstalling it.

5   Tools for the GeoVTag application

In addition to the GeoVTag application itself, three tools have been developed. The first is a GPS simulator. The user enters data like the current time or the current position, and the application builds and sends via Bluetooth the corresponding NMEA sentences. Although it works also on a physical device, this soft is mainly used to test the GeoVTag application on the emulator.

The second tool is a remote tag server. It is a servlet that is able to record new tags and to return all the ones that are available within a certain radius around a given position.

The third and last tool is a program that converts coordinates given in the WGS84 format to the CH1903 format. This allows printing the result of a track on a Swiss

map, in order to observe the quality of the positions given by the GPS. An example is given in Figure 13.
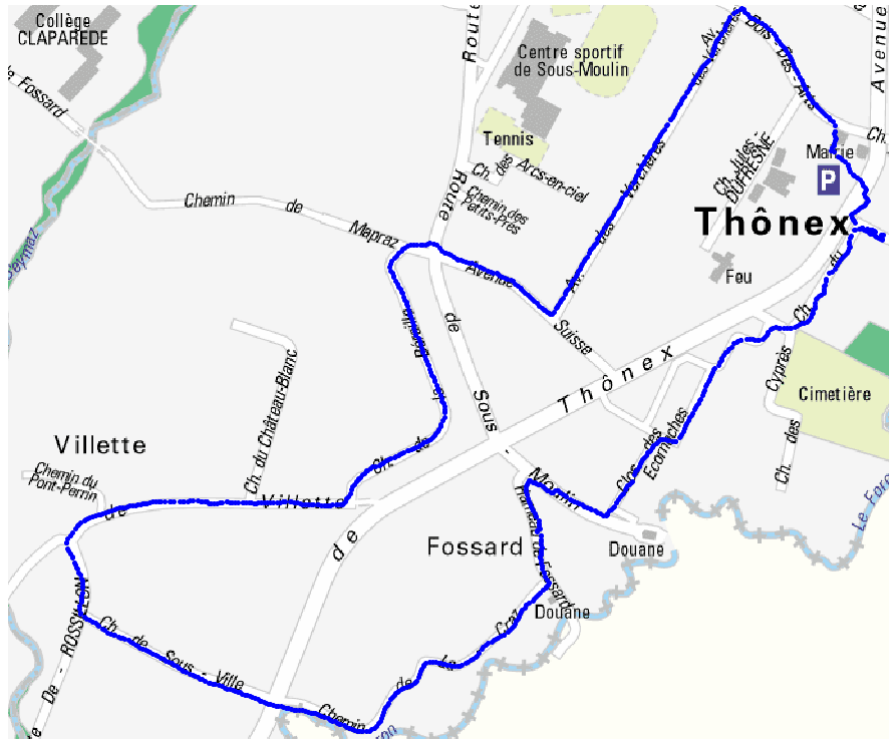

Figure 13

6 Evaluation

We tested the application several times in the field and were quite satisfied about the results. But it consists only in technical tests, to make sure that all the functionalities work properly and that the user interface allows to achieve the current operations easily. We tested the application in the mountains by tracing virtual paths and finding easily some POIs (points of interests), but all the test that have been made so far were done by a single user. What misses in our evaluation is the social aspect, or to observe if and how people would publish tags for other people.

7 Conclusion

This paper presented the GeoVTag application. Although that several in-the-field tests showed us that it can be used in many situations, its main goal is to provide a platform

for further developments. Our idea is to develop a trust and security model for virtual tags. In our Mt-Blanc mountain guides example, the security aspect will ensure that the posted messages are really posted by the signer, that no modification of the original text can be made afterwards, that the service is available for everyone that is authorized, and finally that the messages can be read only by authorized people. But this is not sufficient. Even if you can be sure about the identity of an editor, it is useless if you do not know him and therefore cannot trust the content of his message. That is why we are specifically interested in the trust aspect. This application will be used to test our future trust and security models.

8   Bibliography

[1]   Michel Deriaz. Trust and Security for Spatial Messaging, in OSG Technical Report 2006
[2]   Website: http://www.gpsinformation.org/dale/nmea.htm
[3]   Website: http://www.j2mepolish.org/
[4]   Website: http://cui.unige.ch/~deriazm/Softs/GeoVTag/
[5]   Website: http://java.sun.com/products/sjwtoolkit/download-2_3.html