Towards very simple, yet effective on-the-go incident response preplanning: using publicly-available GIS to improve firefighters' traditional approach

Jose M. Nadal-Serrano
Ayuntamiento de Madrid,
jmnadal@ieee.org

ABSTRACT

Incident response preplanning has an increasing importance in today's Fire Brigades incident response. This paper presents some concepts that could be easily applied, supplying the firefighters with a simple, yet reliable tool that can be configured to include data available at the time of resource activation. This early information and the route map to the incident can be of big help for firefighters if presented in a convenient way. Offline (paper) backup solutions and the need for APIs that may be used to exploit geographic data are also discussed. Finally, a proof of concept setup is developed using GoogleMaps™ for the case of the City of Madrid, Spain.

Keywords

GIS for First Responders, preplanning, firefighter, Emergency Response, critical systems, backup systems.

INTRODUCTION AND RELATED WORK

For many decades, firefighters in Madrid, Spain have been inadvertently doing some preplanning when going for a response. They used to sketch maps on cards, one for every street or important place in the city. Later on, firefighters started to make photocopies of the city map, extracting parts of the street guides and arranging them manually so that they had a map of the route they could follow in order to get to the street in question the easiest and fastest possible, taking into account accessibility and other particular problems. This process proved to be easy and effective: those small maps, usually about an A5 in size, combined with the experience and knowledge of the area covered by a particular Fire Station were enough to get to the incident.

Some years after that, other interesting approaches were put into practice: a group of firefighters studied the whole area of influence of each Fire Station, compiling big listings with all the streets, roads and points of interest plus the directions to get there. This was useful but forced the use of both information sources: on the one hand, the cards holding the map excerpts and on the other hand, the listing with the directions. Some time thereafter, firefighters realized that in many parts of the city there were one-way streets and even one-way streets that changed traffic direction at intervals. Hence, the route to, for instance, street numbers 1-100 was very different to the optimal route to street numbers 101-150. A big effort was made to split streets into intervals, so that a single street was treated as so many streets as the number of intervals in which it was divided. A set of strategic or so-called “basic” points was identified throughout the city, so that directions could be given from the nearest basic point to the incident in the format:

`#BASIC POINT X# / 3L / 2R – PASSED PARK / 1L`,

which reads: *“go to the basic point X, turn left on the third street, then the second on the right, pass the park and take the first on the left: you will be on the block you are going to, now search for the number”*

This basic information, along with the rest of the command given to the first responders (vehicle activated, incident address and other information) was printed in an A4 and handed in to the firefighters. This way they could help the firefighter driving the truck to the incident. Optimal routes for fire trucks had been identified and tested, ensuring the safest and easiest access to the incident's location. These were continuously updated by

Reviewing Statement: This paper has been fully double blind peer reviewed.

certain unit.

However, this new approach was not successful: it is much easier to get the whole picture having a look at a map or sketch than reading directions. Directions were useful and well optimized for fire trucks, but lacked a proper way to visualize the data.

Some years thereafter, modern, special-purpose *GPS navigators* where introduced. Navigators received the incident location by means of a text message from the Communications Center over the TETRA data channel. GPS navigators are a big help, but they still do not take into account the size and accessibility problems of fire trucks. This fact and the lack of other resources prevented the traditional system from evolving: *older cards with manual sketches or photocopies are still being in use*.

Meanwhile, the evolution of the city has led to a significant increase of the effort required to maintain the old paper-based card database: new roads and infrastructures have been built causing old paths and routes to disappear.

The increasing change rate imposes more stress onto firefighters, who have to keep response time low (typically below 5 minutes). Besides, they are no longer able to either know by heart the area of influence of their Fire Station or keep the card database up-to-date.

Moreover, the need for response preplanning (even at a basic level) is increasing since there is a growing demand for information by first responders: they need as much information as possible in order to prepare for the incident physically, psychologically and from and organizational point of view. A map of the location of the incident is in these cases of big help, as it shows potential risks close to the incident's location (buildings close-by, public services possibly affected, accessibility, propagation and evacuation problems, etc). So far, pre-incident plans (or preplans) have been written with big accidents or catastrophes in mind, where many resources are needed [1, 2, 3, 4]. Software suites have been developed to help build such plans [2, 3, 4, 5].

The work presented here shows how information publicly available at the Internet can help initial, very basic response preplanning in the form: *Where are we going? How is accessibility? Will we be able to get there? How can we get there?* and anticipate problems such as *propagation*, the use of special vehicles -*ladders for the rescues*, etc-. This initial, real-time preplanning can be helpful if the incident grows and more resources have to be incorporated.

DESIGN CONSTRAINTS AND THE NEED FOR SIMPLICITY

The environment in which firefighters work is very demanding: high stress, very little time to make decisions, usually at night or with adverse climate conditions. In this situation there is a need to respond to an incident, where one or more people can be dead, injured or at risk. Therefore they have to find a way to get a good picture of the incident by just having a look, taking into account the scarce amount of information given by the Communications Center.

Advantages and Disadvantages of the Previous Systems

It can be surprising to see that even if GPS navigators are of common use these days, firefighters still use the old printed card system. In order to design and improve and existing system, it is important to find out why such innovations where not so well embraced.

	Printed A5 Cards	**Text sheets**	**Custom GPS Navigators**
Advantages	- Easy to use - All the information is assimilated in less than 20s - Optimum route for fire trucks (manually drawn) - Further information given about incident location: accessibility	- Real-time information about the incident - Optimum route for fire trucks (automatic directions from a well-known point or “basic point”)	- Detailed real-time navigation - Real-time information about the incident

	- Overall cost		
Disadvantages	- A5 (too small) - One for every street (not too accurate for very long streets) - Manual update of the database	- Cannot actually see the route ("blind" directions) - No data given about incident location: accessibility, etc. - Poor design (visual layout of the information)	- Not optimum routes for fire trucks - Blind driving (the driver does not know where the GPS is taking them) - Fails often in narrow streets or highway - Can't deal with prohibited turns, etc.

Desired Features

The work presented in this paper focuses on the initial part of the incident response, that is, incident location and accessibility for the Fire Brigade (*Where is the incident? How can we get there?*) plus some other important considerations that affect preplanning (*How is accessibility? How is the environment? What if the incident scales up?*). The experience of the last decades shows that graphical data is preferred over textual data.

It would be therefore highly desirable to have some more features:

- A tool that generates a map or a set of maps and other graphical information: at the very least, the route and the incident location need to be seen clearly and easily at first glance (scrolling is not desired).
- The cartography of the city has to be transparent to the final users (that is, the firefighters) so that there is no need to care about map upgrades and city changes.
- The tool should not require high bandwidths and there has to be a fall-back option in the case of low or no network availability.
- It is important to have a backup system (for instance, in the form of a paper database) in case other systems such as GPS, 3G or other mobile broadband networks were not working.
- It would be highly desirable to use open standards and open source software.
- Less than 30 seconds are needed in order to assimilate the basic information.
- Second-level information such as rescue or traffic constraints must be easy to derive and add real-time.

As we can see, the solution needs to be very simple. *Simplicity is our aim*: the availability of basic, yet-useful information is critical. Further information is desirable and welcome but only if it does not interfere with its critical mission: to guide the first responders from their Fire Station to the incident in a safe and reliable way, effectively taking time constraints into account.

There is a bold reason why it is important to keep the "user interface" as simple as possible, even if there are clearly superior solutions from a technology perspective: the user has to trust and feel comfortable with the system. It is possible to do so without extra functions but core functions are critical. An appropriate UI, familiar to the user (firefighter) is needed. Deployment of the solution is very important as well: final users' have to be involved in the process.

GIS FOR FIRST RESPONDERS

Geographic Information Systems (GIS) are of growing relevance in the last years, since they give information higher value by geolocating it. This is of the utmost importance for firefighters and other emergency services. Administrations have been providing citizens with an increasing amount of services relying on GIS, which may range from pinpointing important buildings or important places to other more complex such as real time feeds showing traffic density. Those applications typically let the user show or hide different data layers and aerial and/or satellite pictures. Most of them are based on proprietary technologies, lacking publicly available Application Programming Interfaces (APIs).

Some ongoing work in the region of Madrid, Spain

It is worth stressing the good ongoing work done in the region of Madrid, Spain by Administrations at municipal level [6] and province level [7]. The information they serve is accurate and would be very useful for our purpose. However, they still do not have an API and therefore it is very difficult to build any application on top of their maps at the time of writing.

A *de facto* Standard: GoogleMaps™

Information giant Google, Inc. has seen geolocation as a value added feature for their Internet searches. In that sense, they have managed to develop easy to use APIs so that final users could build applications on top of their basic GoogleMaps™ service. Several features make their application interesting:

- Map upgrade is completely transparent to the final user. Google has agreements with large, specialized firms such as TeleAtlas. Google relies on the geolocation database and maps of those firms and focuses on its business. The final user knows that s/he just have to comply with GoogleMaps™ API and thus does not care about having to upgrade maps. Google and partners will care about it.
- There are two different APIs (see GoogleMaps™ API Documentation in [8]):
    - *StaticMaps*: It is possible to obtain an image (PNG, JPEG or GIF) as a server reply to a query sent to Google. This is very useful, since it is possible to add features to the image such as size and form factor in pixels, markers and even lines and paths. StaticMaps are a great tool to make the cards we referred to in the previous sections and any other offline or non-interactive applications.
    - *DynamicMaps* or regular API: It is the most powerful GoogleMaps™ API, since it allows many other features such as mash-ups, collection of events in real time and many others. It is specially useful for the aggregation of data and its real time evolution (the so-called mash-ups). It can be specially useful in applications where interactivity is particularly relevant.
- GoogleMaps™ uses standard and well-established information technologies. GoogleMaps™ is a web 2.0 application, which makes it possible to easily access GIS data through a web browser. It is also platform and OS independent. Open source software will therefore interact well with it.
- So far it is possible to access this information at no cost.
- GoogleMaps™ is evolving and new functionality is added to the service regularly.

However, it is important to note that a day may come where Google starts charging for their GoogleMaps™ service or even stop providing it. This is the reason why the GoogleMaps™ service and any application built on top of it cannot be critical, since it is not possible to guarantee availability at all circumstances. Dynamic maps can be used if available to build decision-support applications and help on-the-go preplanning only if the service is not discontinued which will be the rule, not the exception. This problem is not only related to GoogleMaps™ but to any application built on the principle of “cloud computing”: the intelligence is out of reach and therefore it is not controllable.

Finally, it is worth pointing out that it is compulsory to comply with GoogleMaps™ Terms of Service at any time, which can be seen at [9] and whose current version at the time of writing is the one of April 16, 2007. It is important to highlight that the method described here is displayed only as a proof of concept. **Implementing and deploying it would require prior approval by Google, Inc.**

Using Google StaticMaps API for Preplanning Purposes

Leaving aside the huge potential of dynamic javascript maps, we will focus on StaticMaps. Although they do not have the possibilities of their dynamic counterpart, they really fit into the application we are looking for. It is possible to answer our basic preplanning questions by integrating all the preliminary data in an A4 paper.

Two possible scenarios are hereby discussed:

1. The generation of real-time A4 cards, containing all the information coming from the communication center and one or more maps depending on the relative distance from the FireStation to the incident. It would be possible to feed optimized route data to GoogleMaps™ from other sources (such as the database generated in Madrid by the Fire Brigade for internal use).

   This should be the normal mode of operation.

2. The generation of backup A4 cards, one for each street or point of interest containing one or more maps depending on the relative distance to the incident as in the previous case and some space to write notes, either hand-notes or other semi automated ones. These cards would be meant to replace the old A5 cards with newer, bigger and full color ones. They have the advantage that it is possible to update them on a regular basis at almost no additional cost (in contrast to the big effort made to update and redo traditional cards, which had to be updated manually). This is in fact the biggest advantage of this system, effort is big at the beginning, but it rapidly decreases once the first group of cards is made.

   These generated cards would be used as a backup solution, in case real-time generation was not available for whatever reason.

Cards made in the first case provide the firefighters with information that answers these basic questions: *Where is the incident? How do we get there? What happened? Will there be accessibility problems? How can we set cold, warm and hot zones? Etc.* This cards would contain more precise and accurate information than the old ones, since they will be generated in real time (time needed is about 20 seconds for each card in our test setup) including the actual information of the incident. In the second case the card will only contain generic information, since there will be only one card available for each street and thus they will only be used as backups in case it were not possible to generate real time cards. Backup cards would only be able to answer more basic questions such as: *Where is approximately the incident? How do we get there? May there be accessibility problems?.*

PROOF OF CONCEPT: GENERATION OF BACKUP CARDS

With the procedure being described here, it is possible to generate the Backup Cards in Portable Document Format (PDF) for the zone of influence of a Fire Station in the City of Madrid (~2500 streets) in at most three or four days, taking into account the limitations imposed by Google.

Design and Accesibility Issues

It is important to consider other issues that account for usability. These include having clear names, colors, clearly marked starting and final points and optimal routes. Since the cards have to be read and used also at late hours, font type and size and map size has to be carefully chosen so that also older firefighters can use them under any condition.

As mentioned before, there will also be an area containing the name of the street, its number and some place to write annotations. On the backside, two additional maps at a lower zoom level will be added if the street is far from the Fire Station, as well as an environment map. See Figure 1 for details.

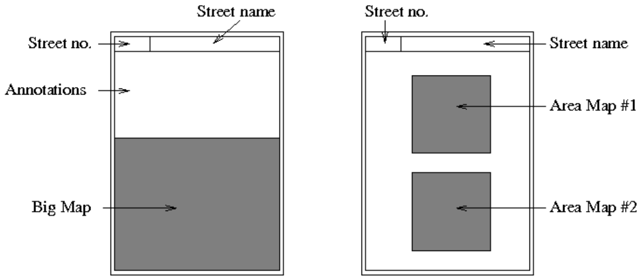

Figure 1. Backup Card Design

Software Requirements and Architecture

The work presented here has been developed using `perl` and *Debian GNU/Linux*, plus some perl libraries needed to access GoogleMaps™ data (`libwww-perl`, `liburi-perl`, etc.). In order to work with the images obtained, a manipulation suite called `imagemagick` is used. *LaTeX* is used for the creation and formatting of the final PDF file. It would be possible to port the setup to other OSs, using perl ports or CygWin.

The architecture of the script is shown in Figure 2.

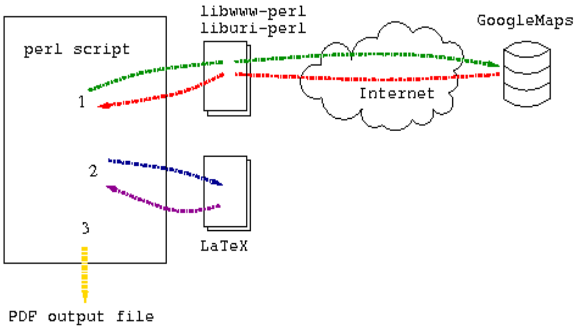

Figure 2. Architecture and data flow

The Script

The script used to generate the backup cards has been designed very modular, so that changes in functionality and scaling can be tackled easily and relatively fast. It includes several options and modes:

- Fire Station number, with the built-in GPS coordinates for all the Fire stations in the City of Madrid.
- Street name, so that it appears at the very top of the card.
- Explicit latitude and longitude, bypassing geocoding.
- Route description
- Annotations
- Route
- Other options for special modes and purposes.

The script works as follows:

1. It reads the command line and parses it, learning about the options desired.
2. Once the line is parsed, a proper query is sent to Google.
3. The response is read, extracting the latitude and longitude of the point in question.
4. Once the coordinates are available, it is possible to ask for the maps. We found that the best trade-off pixel size for the main map was *850x850*px. Since it is not possible to ask for such a big map it was necessary to split that map into four smaller *425x425*px pieces which result in a big area covered while maintaining an acceptable letter size.
5. The relationship between coordinates and pixels is included here but not the whole derivation, which is omitted for the shake of clarity:

```
Longitude: lng_offset = $pixel_width*360/(256*2^$zoom)

Latitude:      a=10^((-4*pi*$no_pixel)/(256*2^$zoom))

b=(1-((1-sin(pi*lat_old/180))/(1+sin(pi*lat_old/180))))/(1+((1-sin(pi*lat_old/180))(1+sin(pi*lat_old/180))))

lat_new=180/pi*asin((1-b*a)/(1+b*a));

lat_offset = lat_new-lat_old;
```

6. Once that relationship is known (for the zoom level we are using, usually 17), the other three pieces of the map are requested. It is usually not desirable to show the end point in the middle of the map we are generating but show more map depending on the direction we are coming from: out of the four tiles, we place the end point on top if we are coming from the south and *viceversa*, and left if we are coming from the east and *viceversa*. See Figure 3.

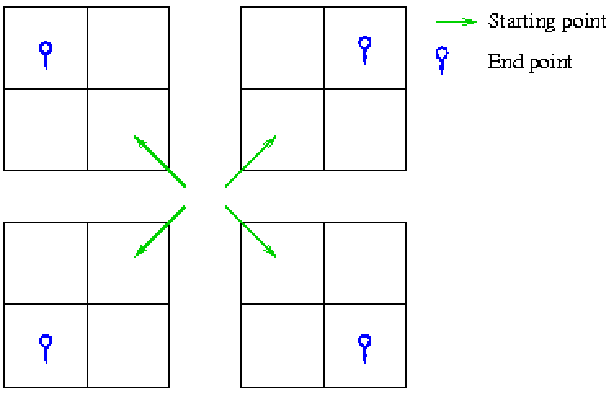

Figure 3. Big map generation depending on the end starting point and end point

7. After that, one or two smaller additional maps are requested if the end point is far or very far away, which help the firefighters know about the environment and the distance to go. It also helps the driver get an idea of where it is if the big map does not provide with enough information about the incident location.
8. The PDF is finally generated using the maps obtained and the *LaTeX* template.

One of the cards obtained using this method is presented in Figure 4.:

Figure 4. Example card obtained with the procedure explained in this paper

FUTURE WORK

The work presented here is currently under evaluation at the City of Madrid, including possible Terms of Use to be enforced by Google.

Further steps would include:

- Generate substitutes for the old A5 printed cards for one of the Fire Stations in order to assess and validate the system.
- Once the formatting and migration procedure is clear, other Fire Stations and eventually all would follow.
- The last step would be to implement the real-time generation of the cards, including incident data, optimized routes for fire trucks and the interface with the incident management system in use.

CONCLUSIONS

Incident response has to be fast but there is always a growing demand for higher quality response standards. Preplanning is of great help in solving these issues, and so it is of growing importance for Fire Brigades. We have hereby presented a *concept and a method* that takes advantage of old –and usually inadvertent practices of preplanning at the Fire Brigade of the City of Madrid. Both aim at improving the existing system by using Information Technologies and taking into account that:

- It is possible to use non-critical facilities and applications to help build backup for critical systems by combining old and new technologies and procedures.
- It is important to weight in non-technical issues, such as accessibility: if the new card system is to be successful, everyone must be able to profit from it. Besides, other issues such as migration procedures have to be carefully studied. Final users are to be taken into consideration through the whole process, constantly incorporating feedback from them.
- A common, modular framework is highly desirable due to its familiar and consistent data visualization for both normal and backup operations. In addition, it provides with a clear interface that can be used to interact with other systems (get further information from the Communications Center about the response, etc.)
- Simplicity, ease of use, display of only relevant information and in some cases the use of non high-end technologies for presentation (i.e. paper instead of electronic displays) can be an advantage when dealing with emergencies.

- For firefighters, extra functionality and cutting-edge technology can be sacrificed for reliability, but not the other way round.
- The use of open source software and de-facto standards such as GoogleMaps™, plus well-defined programming interfaces open the door to further improvement and adaptation to other cities and Fire Brigades, which can easily adapt certain implementation to their particular needs. It is worth pointing out that the proposed back-up solution could be seen as a cheap replacement for GPS navigation devices in developing countries.

ACKNOWLEDGEMENTS

To all the people that have supported me during this time, both inside and outside the fire Brigade.

REFERENCES

1. NFPA 1620: Recommended Practice for Pre-Incident Planning, 2003 Edition, NFPA
2. SAR Technology: 'Incident Commander Pro' Software, *http://sartechnology.ca/sartechnology/ST_TerroristResponse.htm*
3. Fireplanning Associates, *http://www.fireplanningassociates.com/*
4. VIPlanning Software, *http://www.viplanning.com/Content/preincidentplan.htm*
5. Firemapping, *http://firemapping.wordpress.com/2008/10/30/how-to-symbolize-preplan/*
6. Callejero de Madrid Online, Ayuntamiento de Madrid, *http://www-1.munimadrid.es/guia/visualizador/login.do*.
7. Nomecalles, Callejero de la Comunidad de Madrid, *http://gestiona.madrid.org/nomecalles/Inicio.icm*
8. Google Maps API Documentation, *http://code.google.com/apis/maps/documentation/*
9. GoogleMaps™ Terms of Service, *http://code.google.com/apis/maps/terms.html*

APPENDIX: LATITUDE AND LONGITUDE COMPUTATION

In the official documentation provided by Google for their GoogleMaps™ API [3] it is possible to find that:

$$absolute\_X\_pixel\_number = \frac{(longitude + 180)\cdot 256 \cdot 2^{zoom}}{360}$$, and

$$absolute\_Y\_pixel\_number = \frac{256\cdot 2^{zoom}}{2} - \frac{0.5\log\left(\frac{1+\sin\left(\frac{latitude\cdot\pi}{180}\right)}{1-\sin\left(\frac{latitude\cdot\pi}{180}\right)}\right)\cdot(256\cdot 2^{zoom})}{2\pi},$$

where `absolute_X_pixel_number` and `absolute_Y_pixel_number` represent the absolute pixel number measured from longitude and latitude 0. `zoom` represents the zoom level, which goes up to 17. Both of them take into account that Google uses tiles *256x256*px in size.

Hence, if we are only deriving the change in latitude and/or longitude for a certain pixel size, we have:

$$\Delta X_{pixel} = \frac{\Delta longitude \cdot 256\cdot 2^{zoom}}{360}$$, that is: $$\Delta longitude = \frac{\Delta X_{pixel}\cdot 360}{256\cdot 2^{zoom}}$$

Correspondingly, for the latitude we have that:

$$Y_{pixel} = \frac{256\cdot 2^{zoom}}{2} - \frac{0.5\log\left(\frac{1+\sin\left(\frac{latitude\cdot\pi}{180}\right)}{1-\sin\left(\frac{latitude\cdot\pi}{180}\right)}\right)\cdot(256\cdot 2^{zoom})}{2\pi}$$

Solving for latitude we have that:

$$\Delta latitude = \frac{180}{\pi}\arcsin\left(\frac{1-\frac{1-\frac{\pi}{180}\cdot old_{latitude}}{1+\frac{\pi}{180}\cdot old_{latitude}}\cdot 10^{\frac{-4\pi\Delta Y_{pixel}}{256\cdot 2^{zoom}}}}{1+\frac{1-\frac{\pi}{180}\cdot old_{latitude}}{1+\frac{\pi}{180}\cdot old_{latitude}}\cdot 10^{\frac{-4\pi\Delta Y_{pixel}}{256\cdot 2^{zoom}}}}\right) - old_{latitude}$$