

FIXIT: Ushahidi location lists

Today, a conversation amongst crismappers went something like: "when are we going to migrate to IRC?" (this in response to yet another Skype idiosyncrasy getting in the way of the team), then "not until we have a decent interface for it on all platforms" and "we asked the weekend hackathons for this, but it's not sexy: you can't tell people you're saving the world, or helping starving children with it". That's a whole pile of cynicism and frustration, but behind it are three things: 1) mappers still don't have all the tools they need, and are relying on people-as-processors to get round that, 2) mappers don't know how to ask for tools in ways that gets them what they need, and 3) hackathons may not be the best place to get those non-sexy tools built.

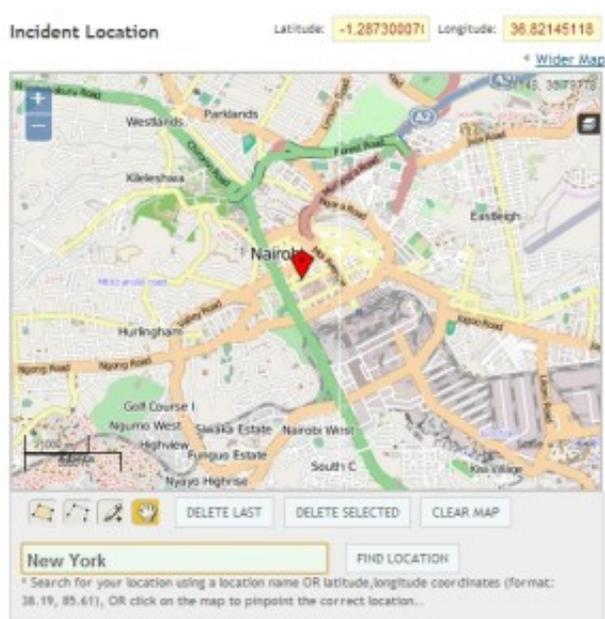
So. Options. Best is to find a good piece of existing open-source kit that almost meets mappers' needs, and extend it with the things needed. Less-good is to build all-new kit - mappers aren't hackers, and new kit inevitably breaks and needs maintaining and training. In-between is adding things onto proprietary kit using APIs (if they don't get removed - yes, I'm looking at you, Skype team), and adapting existing open-source kit that doesn't meet needs, but could be adapted (with the programming equivalent of a carefully-swung sledgehammer). But that's just options from the mapping teams' perspective: another option is to team up with a coding company that wants to build tools for a similar (or adjacent) market.

I'm just as guilty for not documenting the things that I think need improving. I'm used to writing "FIXIT" in my code when I know there's a potential problem in it - so here's the start of a set of posts about things that could be upgraded. Some of them, I'll start trying to fix - and document that in updates to each FIXIT blogpost.

OverCognition

Journeys through development data.

<http://overcognition.com>



superseded

- When you press the "find location" button, Ushahidi calls (eventually) a function called `geocode($address)` in file `application/helpers/map.php`
- This calls the google map api: for instance, for the address "Mount Arlington", it calls <http://maps.google.com/maps/api/geocode/json?sensor=false&address=Mount%20Arlington> (don't worry about the %20: it's the internet's way of saying "space"). The API call produces json data (go ahead and look at the page - it's all there), which Ushahidi then pulls the country, country id, location name, latitude and longitude from.
- Erm. That's it.

A geolocation team on a deployment is responsible for finding lat/longs for location names. They usually keep a list of these lat/longs and location names, and that's usually in a Google spreadsheet. So what we need here is:

- An Ushahidi variable holding the address of the geolocators' spreadsheet.
- That spreadsheet to be in a recognisable form - e.g. it has a column each for latitude, longitude and placename
- A piece of code inserted into function `geocode($address)`, that when the Google map API comes up blank, checks the Google spreadsheet for that location instead (or maybe checks the spreadsheet first: that depends on how the geolocation teams usually work). That piece of code will need to use the [googledocs API](#), which is possibly the hardest part of this plan.

- Maybe even (shock, horror), check the *other* map APIs (openstreetmap etc) too.

None of this is horrendous, but it does take time to do. Perhaps someone somewhere will think that it's worth it.