

Project idea: visualising crismapping categories

I'm a [crismapper](#). I've seen or worked on most crisis [Ushahidi](#) maps since January 2010, and I've watched the categories used on them split and evolve over time (from Haiti's emergencies and public health to Snowmageddon's problems / solutions and beyond).

Cat from [Humanity Road](#) has kept a screenshot of the categories on each of the main deployments since then - when I saw it today, it reminded me immediately of work that Global Pulse did on visualising category evolution across news articles, with articles as nodes connected by subject, and coloured by main category, as discovered by gisting the articles.

Except this time, we don't have to guess the categories (although doing that later by mining text in the reports in each category could be fun). Each Ushahidi map comes with a set of categories - each report (piece of geolocated information) is tagged by the categories it belongs to.

So. The simple version of the visualisation (and I'm thinking that [Processing](#) should be able to handle much of this). The y-axis in the visualisation is a timeline. Each deployment is a column of categories, represented as graph nodes: we then put lines from deployment(n) category(x) to deployment(n+1) category(y) if they're related.

Working out relations between categories isn't totally trivial. Language differences (i.e. arabic to english), generalisation/specialisation and synonyms do happen, so we might need to build a category ontology to handle this. Line weights could be used to show the strength of connection between pairs of categories. Ushahidi also uses sub-categories on some deployments, so we'll have to work out how to group these - perhaps by colour, perhaps by putting a box around them.

Making it pretty: we can use some standard tools (like matrix [normalisation](#)) to minimise the number of crossed lines between deployments.

So you'd have a meta-graph where the nodes are deployments, and inside each of those you'd have supercategories and categories. One variant of this is to visualise all the deployments as metanodes in free space, i.e. scattered all over the place, and place the strongest-related deployments (by categories) closest to each other. I don't know if there are packages that allow drilling into metanodes, i.e. to click on a metanode and view all the links from its categories out to other metanodes, but that could be fun too.

Once we have the connectors (e.g. into Ushahidi's publically-accessible reports on [CrowdMap](#)), all sorts of other cool visualisations suddenly become possible too. But finally I need to ask the zeroth question about a project, i.e. "what is the problem we're trying to solve here" (first question is "is there any value in it"). I think what I'm trying to do is make an easy way for crismapping historians to see how deployments have evolved, and give crismapping leads an easily-visualised example set of categories that worked for different deployments over time. The value might be

OverCognition

Journeys through development data.

<http://overcognition.com>

new category sets, or an awareness of arcs that the category types are currently on. I don't know. But I'm sure that I know several thousand people out there with opinions about it...