

Strata talk on hunchworks technology

I try not to put too much dayjob stuff here, but sometimes I need to leave less-tidy breadcrumbs for myself. Here's the 10-minute (ish) talk I gave at Strata New York this year.

Intro

I'm Sara Farmer, and I'm responsible for technology at Global Pulse. This brings its own special issues. One thing we're passionate about is **making systems available that help the world**. And one thing we've learnt is that we **can't do that without cooperation both within and outside the UN**. We're here to make sure **analysts and field staff get the tools that they need**, and that's a) a **lot of system** that's needed, and b) something that **organisations across the whole humanitarian, development and open data space need too**.

We're **talking to those other organisations about what's needed, and how we can best pool our resources to build the things that we all need**. And the easiest way for us to do that is to **release all our work as open-source software, and act as facilitators and connectors** for communities rather than as big-system customers.

Open source isn't a new idea in the UN - **UNDP and UNICEF, amongst others, have trailblazed for us, and we're grateful to have learnt** from their experience in codejams, hackathons and running git repositories and communities. And it's a **community of people like Adaptive Path and open source coders who make HunchWorks happen**, and I'd like to publicly thank their dedication and the dedication of our small tech team. We have **more work to do, not least in building a proper open innovations culture across the UN (we've codenamed this "Blue Hacks")** and **selecting licensing models that allow us to share data and code** whilst meeting the UN's core mandate, but in this pilot project it's working well.

We're **already building out the human parts** of the system (trust, groups etc) but **Hunchworks doesn't work in isolation**: it needs to be integrated into a **wider ecosystem of both existing and new processes, users and technologies**. The humanitarian technology world is changing rapidly, and we've spent a lot of time thinking about **what it's likely to look like both soon and in the very near future**.

So. For hunchworks to succeed, it must **connect to four other system types**.

We need users, and we need to build up our user population quickly. We're talking about **lab managers, development specialists, mappers, policy coordinators, local media and analysts**. Those users will need also **specialist help from other users in areas like maps**,

crowdsourcing, data and tools curation.

- UN Teamworks - this is a Drupal-based content management system that a UNDP New York team has created to **connect groups of UN people with each other, and with experts beyond the UN.**
- Professional networking systems like LinkedIn that **many crisis professionals are using to create networks.**
- Note that **some of our users will be bots** - more on this in a minute.

Users will need **access to data**, or (in the case of data philanthropy), to search results.

- UN SDI - a project creating **standards and gazetteers for geospatial data** across the UN.
- CKAN - **data repository nodes**, both for us and from open data initiatives.
- Geonode, JEarth et al - because **a lot of our data is geospatial.**

They'll need **tools to help them make sense** of that data. And bots to do some of that automatically.

- We need toolboxes - ways to search through tools in the same way that we do already with data. We're talking to people like Civic Commons about the best ways to build these.
- We're building apps and plugins where we have to, but we're talking about organisations putting in nodes around the world, so we're **hunting down open source and openly available tools wherever we can.** We're waiting for our first research projects to finish before we finalise our initial list, but we're going to **at least need data preparation, pattern recognition, text analysis, signal processing, graphs, stats, modelling and visualisation tools.**
- Because we want to send hunchworks instances to the back of beyond, we're also **including tools that could be useful in a disaster** - like Ushahidi, Sahana, OpenStreetMap and Google tools.
- And there are **commercial tools and systems that we're going to need to interface with** too. We're talking about systems like Hunch and a bunch of other suppliers that we'll be **talking to once we get the panic of our first code sprints out of the way.**

And they need a '**next**', **a way to spur action**, to go with the knowledge that Hunchworks creates.

- We're adding tools for this too. And also connecting to **UN project mapping systems:**
- UN CRMAT - risk mapping and project **coordination during regional development**
- UN CIMS - project coordination during humanitarian crises, an **extension of the 3W** (who, what, where) idea.

Which is a big vision to have and a lot to do after our first releases next spring. And yet another reason why we're going to need to do all the partnering and facilitation that we can.

So. You've seen how we've designed Hunchworks to help its users work together on hunches. But Hunchworks is more than just a social system, and there are a lot of algorithms needed to make that difference. We have to design and implement the **algorithms that make Hunchworks smart enough to show its users the information that is relevant to them when they need it** (also known as looking for all the boxes marked "**and then a miracle happens**").

And the first algorithms needs are here:

- **Similarity and complementarity metrics.** We need to work on both of these. Now there's a lot of work out there on how things are similar, but there's **not so much around about how people and their skills can complement each other**. We've been looking at things like robot team theories, autonomy and human-generated team templates as baselines for this.
- **Relevance.** And for that, read "need some interesting search algorithms". We're looking into search, but we're also looking at user profiling and focus of attention theories, including **how to direct users' peripheral attention** onto things that are related to a hunch that they're viewing.
- **Credibility.** We'd like to combine all the information we have about hunches and evidence (including user support) into estimates of belief for each hunch, that we can use as **ratings for hunches, people and evidence sources**. There's work in uncertain reasoning, knowledge fusion and gamification that could be helpful here, and there are some excellent examples already out there on the internet. As part of this, we're also looking at how **Hunchworks can be mapped onto a reasoning system**, with hunches as propositions in that system. Under "everything old is new again", we're interested in how that correlates to 1980s reasoning systems too.
- **Hunch splitting, merging and clustering.** We need to know when hunches are similar enough to suggest merging or clustering them. We also would like to highlight when a hunch description and the evidence attached to it deviates far enough from its original description to consider splitting it into a group of related hunches. Luckily, one of our research projects has addressed exactly this problem - this is an example of how **our internal algorithm needs are often the same as the users' tool needs** - and we're looking into how to adapt it.

Mixing human insight with big data results. One of the things that makes Hunchworks more than just a social system is the way that we **want to handle big data feeds**. We don't think it's **enough to give analysts RSS feeds or access to tools**, and we're often working in environments where **time is our most valuable commodity**. The big question is how we can best combine human knowledge and insight with automated searches and analysis.

Let's go back to Global Pulse's mission. **We need to detect crises as they unfold in real time, and alert people who can investigate them further and take action to mitigate their effects.**

It's **smart for us to use big data tools** to detect 'data exhaust' from crises. It's **smart for us to add human expertise** to hunches that something might be happening. But it's **smarter still for us to combine these two information and knowledge sources** into something much more powerful.

We've argued a lot about how to do this, but the arguments all seem to boil down to one question: "**do we treat big data tools as evidence or users in Hunchworks**"? If we treat big data tools as evidence, we have a relatively easy life - we can rely on users to use the tools to generate data that they attach to hunches, or can set up tools to add evidence to hunches based on hunch keywords etc. But the more we talked about what we wanted to do with the tools, from being able to create hunches automatically from search results to rating each tool's effectiveness on a given type of hunch, **the more they started sounding like users**.

So we've decided to use **bots. Agents. Intelligent agents**. Whatever you personally call a piece of code that's wrapped in something that observes its environment and acts based on those observations, **we're treating them as a special type of Hunchworks user**. And by doing that, we've given bots the **right to post hunches** when they spot interesting patterns; the **ability to be rated** on their results, and the **ability to be useful members of human teams**.

And now I'll start talking about the things that are difficult for us. You've already seen that trust is incredibly important in the Hunchworks design. Whilst we have to build the system to enhance trust between users and users, users and bot, users and the system etc, we also have to build to **deal with what happens when that trust is broken**. Yes, we need security.

We need security to ensure that hidden hunches are properly distanced from the rest of the system. I've worked responses where people died because they were blogging information, and we need to minimise that risk where we can.

We also need ways to spot sock puppet attacks and trace their effects through the system when they happen. This is on our roadmap for next year.

And then we have **localisation**. The UN has 6 official **languages** (arabic, chinese, english, french, russian and spanish), but we're going to be putting labs into countries all over the world, each with their own languages, data sources and cultural styles. We can't afford to lose people because we didn't listen to what they needed, and this is a lot of what the techs embedded in Pulse Labs will be doing. We'll need help with that too.

Also **on the 'later' roadmap is dealing with federation**. We're starting with a single instance of Hunchworks so we can get the user management, integration and algorithm connections right. But we're part of an organisation that works across the world, including places where bandwidth is

very limited and sometimes non-existent, and mobile phones are more ubiquitous than computers. We're also working out ways, as part of Data Philanthropy, to connect to appliances embedded within organisations that can't, or shouldn't, share raw data with us. Which means a federated system of platforms, with all the synchronisation, timing and interface issues that entails.

We aren't addressing federation yet, but we are learning a lot in the meantime about its issues and potential solutions from systems like **Git** and from our interactions with the **crisismapper communities** and other organisations with similar operating structures and connectivity problems. Crisismapping, for example, is teaching us a lot about how people handle information across cultures and damaged connections whilst under time and resource stress.

Okay, I've geeked out enough on you. Back to Chris for the last part of this talk.

Slideset

Focussed on the technical challenges of building Hunchworks... slide titles are:

- Open source and the UN - pic of codejam - licensing models, working models for UN technology creation (h/t unicef and undp, big h/t adaptive path and the codejammers)
- Integration - bubble diagram with the other UN systems that we're connecting to (CRMA and CIMS for actions, TeamWorks for the user base, non-UN systems for tools - CKAN for data, CivicCommons for toolbag etc)
- Algorithms - list of algorithms we're designing, discussion of uncertainty handling, risk management, handling mixed human-bot teams, similarity and complementarity metrics (including using hand-built and learnt team templates) etc
- Security - how to handle hidden hunches, incursions and infiltrations. Diagram of spreading infiltration tracked across users, hunches and evidences.
- Localisation - wordcloud of world languages; discuss languages and the use of pulse labs
- Federation - clean version of my federation diagram - describe first builds as non-federated, but final builds being targeted at a federated system-of-systems, with some nodes having little or no bandwidth and the synchronisation and interface needs that creates